



LQG/LTR Tilt and Tip Control for the Starfire Optical Range 3.5-meter
Telescope's Adaptive Optics System

THESIS

Neil D. Paris, Captain, USAF

AFIT/GE/ENG/06-45

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or U.S. Government.

AFIT/GE/ENG/06-45

LQG/LTR Tilt and Tip Control for the Starfire Optical Range 3.5-meter
Telescope's Adaptive Optics System

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Neil D. Paris, B.S.E.E.
Captain, USAF

March 2006

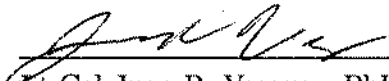
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

LQG/LTR Tilt and Tip Control for the Starfire Optical Range 3.5-meter
Telescope's Adaptive Optics System

Neil D. Paris, B.S.E.E.

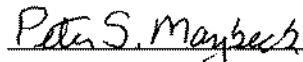
Captain, USAF

Approved:


Lt Col Juan R. Vasquez, PhD
Thesis Advisor

28 Feb 06

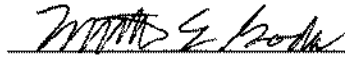
Date



Dr. Peter S. Maybeck
Committee Member

28 Feb 06

Date



Lt Col Matthew E. Goda, PhD
Committee Member

28 Feb 06

Date



Dr. James M. Brown
Committee Member

28 Feb 06

Date

Table of Contents

	Page
List of Figures	vi
List of Tables	xiv
List of Abbreviations	xv
Abstract	xvi
Acknowledgments	xvii
 I. Introduction	 1-1
1.1 Problem Statement	1-1
1.2 Scope of the Research	1-2
1.3 Research Contributions	1-4
1.4 Thesis Overview	1-4
 II. Background	 2-1
2.1 Atmospheric Phase Distortion	2-1
2.2 Adaptive Optics	2-3
2.2.1 General Description	2-3
2.2.2 Starfire Optical Range 3.5-meter Telescope	2-4
2.3 Optimal Control Techniques	2-7
2.3.1 Optimal Control Description	2-7
2.3.2 Linear Quadratic Gaussian Control with Loop Trans- fer Recovery	2-7
2.3.3 H_∞	2-9
2.3.4 Implemented Controller Design Technique	2-10

	Page
III. Theory and Controller Design	3-1
3.1 Atmospheric Turbulence Theory	3-1
3.1.1 Atmospheric Turbulence Strength	3-1
3.1.2 Zernike Polynomials	3-2
3.1.3 Greenwood Frequency	3-3
3.2 LQG Theory	3-4
3.2.1 LQR	3-4
3.2.2 Kalman Filter	3-10
3.2.3 LQG/LTR	3-13
3.2.4 Summary	3-16
3.3 Controller Design	3-17
3.3.1 Design Parameters	3-19
3.3.2 Kalman Filter Model	3-19
3.3.3 LQG/LTR Tracker Design	3-35
3.4 Summary	3-40
IV. Simulation and Analysis	4-1
4.1 Simulation	4-1
4.1.1 Tilt and Tip Simulation	4-1
4.1.2 Control Loop Simulation	4-10
4.2 Analysis of Simulation Results	4-19
4.2.1 Baseline Cases	4-21
4.2.2 Sensor Noise Cases	4-68
4.2.3 Additional Baseline Case	4-111
4.3 Summary	4-112

	Page
V. Conclusions and Recommendations	5-1
5.1 Summary	5-1
5.2 Conclusions	5-2
5.3 Recommendations	5-5
Appendix A. System Model Matrices	A-1
A.1 Continuous-Time Plant State-Space Matrices	A-1
A.1.1 Modified Jordan Canonical Form	A-1
A.2 Kalman Filter Model State-Space Matrices	A-7
A.2.1 Equivalent Discrete-Time	A-7
A.3 LQR Model State-Space Matrices	A-13
A.3.1 Equivalent Discrete-Time	A-13
A.4 LTR-Based State Weighting Matrices	A-20
Appendix B. MATLAB [®] Code	B-1
B.1 Tilt and Tip Simulation	B-1
B.2 Pyramid Sensor Noise Characterization	B-9
B.3 Controller Design	B-13
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure		Page
2.1.	Effect of Atmospheric Turbulence	2-2
2.2.	Non-disturbed vs. Disturbed Wavefront	2-2
2.3.	SOR 3.5-meter Telescope	2-4
2.4.	Optical Relay Path	2-5
3.1.	Control Loop	3-18
3.2.	Optical Path Through the Steering Mirrors	3-18
3.3.	CSM Frequency Response Data	3-20
3.4.	Provided CSM Azimuth Model versus the Frequency Response Data	3-21
3.5.	Provided CSM Elevation Model versus the Frequency Response Data	3-21
3.6.	FSM Frequency Response Data	3-22
3.7.	Provided FSM Azimuth Model versus the Frequency Response Data	3-23
3.8.	Provided FSM Elevation Model versus the Frequency Response Data	3-23
3.9.	Azimuth Model #2	3-25
3.10.	Azimuth Model #3	3-26
3.11.	Azimuth Model #9	3-26
3.12.	Elevation Model #2	3-27
3.13.	Elevation Model #3	3-27
3.14.	Elevation Model #4	3-28
3.15.	Example of determining the angular tilt using 2π maximum phase .	3-33
4.1.	Simplified Example of Two-Dimensional PSD Coordinate Frames .	4-3
4.2.	Scenario 1 Tilt and Tip Values	4-6
4.3.	Scenario 2 Tilt and Tip Values	4-6
4.4.	Scenario 3 Tilt and Tip Values	4-7
4.5.	Scenario 4 Tilt and Tip Values	4-7

Figure		Page
4.6.	Scenario 1 Structure Function Comparison	4-8
4.7.	Scenario 2 Structure Function Comparison	4-8
4.8.	Scenario 3 Structure Function Comparison	4-9
4.9.	Scenario 4 Structure Function Comparison	4-9
4.10.	Simulink [®] Model	4-11
4.11.	Quad Cell Depiction	4-12
4.12.	Pyramid Sensor Azimuth Axis Output Characteristics	4-13
4.13.	Pyramid Sensor Elevation Axis Output Characteristics	4-13
4.14.	Scenario 1 Shifted Tilt and Tip Values	4-17
4.15.	Scenario 2 Shifted Tilt and Tip Values	4-18
4.16.	Scenario 3 Shifted Tilt and Tip Values	4-18
4.17.	Scenario 4 Shifted Tilt and Tip Values	4-19
4.18.	Case (1a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-24
4.19.	Case (1b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-25
4.20.	Case (1c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-25
4.21.	Case (2a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-26
4.22.	Case (2b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-26
4.23.	Case (2c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-27
4.24.	Case (1a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-27
4.25.	Case (1b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-28
4.26.	Case (1c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-28
4.27.	Case (2a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-29
4.28.	Case (2b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-29
4.29.	Case (2c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-30
4.30.	Case (3a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-31

Figure		Page
4.31.	Case (3b): $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-31
4.32.	Case (3c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-32
4.33.	Case (4a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-32
4.34.	Case (4b): $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-33
4.35.	Case (4c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-33
4.36.	Case (6a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-35
4.37.	Case (6b): $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-35
4.38.	Case (6c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-36
4.39.	Case (6a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-36
4.40.	Case (6b): $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-37
4.41.	Case (6c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-37
4.42.	Case (7.1a): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-38
4.43.	Case (7.1b): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-38
4.44.	Case (7.1c): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-39
4.45.	Case (8.1a): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-39
4.46.	Case (8.1b): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-40
4.47.	Case (8.1c): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-40
4.48.	Case (7.4a): Delay 4, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-41
4.49.	Case (7.4a): Delay 4, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-42
4.50.	Case (7.5b): Delay 5, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-43
4.51.	Case (7.5c): Delay 5, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-43
4.52.	Case (7.9b): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-44
4.53.	Case (7.9b): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-44
4.54.	Case (7.9c): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-45

Figure		Page
4.55.	Case (7.9c): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-46
4.56.	Case (8.9c): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-46
4.57.	Case (8.9c): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-47
4.58.	Case (7.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-48
4.59.	Case (7.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-48
4.60.	Case (8.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-49
4.61.	Case (8.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-49
4.62.	Case (10.1a): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-51
4.63.	Case (10.1b): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-51
4.64.	Case (10.1c): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-52
4.65.	Case (10.1a): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-52
4.66.	Case (10.1b): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-53
4.67.	Case (10.1c): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-53
4.68.	Case (10.3a): Delay 3, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-54
4.69.	Case (8.3a): Delay 3, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-55
4.70.	Case (10.8b): Delay 8, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-55
4.71.	Case (8.8b): Delay 8, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-56
4.72.	Case (10.9c): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-56
4.73.	Case (10.4a): Delay 4, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-57
4.74.	Case (10.9b): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-58
4.75.	Case (10.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-58
4.76.	Case (11.4a): Delay 4, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-60
4.77.	Case (12.4a): Delay 4, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-60
4.78.	Case (11.9b): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-61

Figure		Page
4.79.	Case (12.9b): Delay 9, $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-61
4.80.	Case (11.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-62
4.81.	Case (12.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-62
4.82.	Case (13a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-63
4.83.	Case (14.a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-64
4.84.	Case (13b): $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-64
4.85.	Case (14b): $\mathbf{U}_{diag} = \begin{bmatrix} 3E5 & 3E5 & 1E4 & 1E4 \end{bmatrix}$	4-65
4.86.	Case (13c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-65
4.87.	Case (14c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-66
4.88.	Case (1a): Steering Mirror Outputs (Azimuth Axis)	4-66
4.89.	Case (1b): Steering Mirror Outputs (Azimuth Axis)	4-67
4.90.	Case (1c): Steering Mirror Outputs (Azimuth Axis)	4-67
4.91.	Case (15a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-71
4.92.	Case (15b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-72
4.93.	Case (15c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-72
4.94.	Case (16a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-73
4.95.	Case (16b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-73
4.96.	Case (16c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-74
4.97.	Case (15a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-74
4.98.	Case (15b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-75
4.99.	Case (15c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-75
4.100.	Case (16a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-76
4.101.	Case (16b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-76
4.102.	Case (16c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-77

Figure		Page
4.103.	Case (17.1a): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix} \dots$	4-78
4.104.	Case (17.1b): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix} \dots$	4-78
4.105.	Case (17.1c): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix} \dots$	4-79
4.106.	Case (18.1a): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix} \dots$	4-79
4.107.	Case (18.1b): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix} \dots$	4-80
4.108.	Case (18.1c): Delay 1, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix} \dots$	4-80
4.109.	Case (17.5a): Delay 5, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix} \dots$	4-82
4.110.	Case (17.5b): Delay 5, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix} \dots$	4-82
4.111.	Case (17.5c): Delay 5, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix} \dots$	4-83
4.112.	Case (17.10a): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix} \dots$	4-84
4.113.	Case (17.10b): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix} \dots$	4-84
4.114.	Case (17.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix} \dots$	4-85
4.115.	Case (18.10a): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix} \dots$	4-85
4.116.	Case (18.10b): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix} \dots$	4-86
4.117.	Case (18.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix} \dots$	4-86
4.118.	Case (17.10a): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix} \dots$	4-87
4.119.	Case (17.10b): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix} \dots$	4-87
4.120.	Case (17.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix} \dots$	4-88
4.121.	Case (18.10a): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix} \dots$	4-88
4.122.	Case (18.10b): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix} \dots$	4-89
4.123.	Case (18.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix} \dots$	4-89
4.124.	Case (19.10a): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix} \dots$	4-90
4.125.	Case (19.10b): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix} \dots$	4-91
4.126.	Case (19.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix} \dots$	4-91

Figure		Page
4.127.	Case (20.10a): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-92
4.128.	Case (20.10b): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-92
4.129.	Case (20.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-93
4.130.	Case (21a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-94
4.131.	Case (21b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-94
4.132.	Case (21c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-95
4.133.	Case (22a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-95
4.134.	Case (22b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-96
4.135.	Case (22c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-96
4.136.	Case (23.5a): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$. .	4-97
4.137.	Case (23.5a): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$. .	4-98
4.138.	Case (23.5b): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-98
4.139.	Case (23.5b): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-99
4.140.	Case (23.5c): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-99
4.141.	Case (23.5c): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-100
4.142.	Case (24.5a): Delay 5, $f_G = 189 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$.	4-101
4.143.	Case (24.5a): Delay 5, $f_G = 189 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$.	4-101
4.144.	Case (24.5b): Delay 5, $f_G = 189 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-102
4.145.	Case (24.5b): Delay 5, $f_G = 189 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-102
4.146.	Case (24.5c): Delay 5, $f_G = 189 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-103
4.147.	Case (24.5c): Delay 5, $f_G = 189 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-103
4.148.	Case (25.5a): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$. .	4-104
4.149.	Case (25.5a): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$. .	4-105
4.150.	Case (25.5b): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-105

Figure		Page
4.151.	Case (25.5b): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-106
4.152.	Case (25.5c): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-106
4.153.	Case (25.5c): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-107
4.154.	Case (26a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-107
4.155.	Case (26b): $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-108
4.156.	Case (26c): $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-108
4.157.	Case (15a): Steering Mirror Outputs (Azimuth Axis)	4-109
4.158.	Case (15b): Steering Mirror Outputs (Azimuth Axis)	4-110
4.159.	Case (15c): Steering Mirror Outputs (Azimuth Axis)	4-110
4.160.	Case (27.4a): Delay 4, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-113
4.161.	Case (27.4a): Delay 4, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-113
4.162.	Case (27a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-114
4.163.	Case (27a): $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-114
4.164.	Case (27.10a): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$	4-115
4.165.	Case (27.10b): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$	4-115
4.166.	Case (27.10c): Delay 10, $\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$	4-116

List of Tables

Table		Page
3.1.	Design Parameters	3-19
3.2.	CSM Transfer Function Poles and Zeros	3-20
3.3.	FSM Azimuth Transfer Function Error Statistics	3-24
3.4.	FSM Elevation Error Statistics	3-24
3.5.	FSM Azimuth Axis Poles and Zeros	3-29
3.6.	FSM Elevation Axis Poles and Zeros	3-29
4.1.	Tilt and Tip Simulation Parameters	4-5
4.2.	Tilt and Tip Simulation Scenarios	4-5
4.3.	Sensor Noise Variance	4-14
4.4.	Unbiased Variance of Sensor Noise	4-15
4.5.	Baseline Simulation Cases	4-16
4.6.	Sensor Noise Simulation Cases	4-16
4.7.	Kalman Filter Measurement Noise	4-69
4.8.	Kalman Filter Dynamics Noise	4-70
4.9.	LQG/LTR Performance	4-111

List of Abbreviations

Abbreviation	Page
AFRL Air Force Research Laboratory	2-4
SOR Starfire Optical Range	2-4
WFS Wavefront Sensor	2-5
CCD Charged Coupled Device	2-5
FSM Fine Steering Mirror	2-6
CSM Coarse Steering Mirror	2-6
LQG Linear Quadratic Gaussian	2-7
LQR Linear Quadratic Regulator	2-7
SR Stochastic Regulator	2-8
LTR Loop Transfer Recovery	2-9
H-V Hufnagel-Valley	3-1
PSD Power Spectral Density	3-2
WGN white Gaussian noise	3-8
PI Proportional Integral	3-35

Abstract

The Air Force Research Laboratory has sponsored research on the tracking control loop portion of the adaptive optics system in the Starfire Optical Range 3.5-meter telescope at Kirtland Air Force Base. The control loop includes two steering mirrors (Coarse Steering Mirror and Fine Steering Mirror) used to remove wavefront tilt and tip phase distortion from light entering the telescope. Currently, the control loops for each mirror are independent and operate in parallel without any knowledge of what the other controller and mirror are doing. The objective of this research is to design a single Linear Quadratic Gaussian controller to control both steering mirrors in order to eliminate wavefront tilt and tip distortions induced by the earth's atmosphere, and to evaluate the stability robustness and performance of the controller through simulation and Monte Carlo analysis. Controller design elements and simulation parameters are varied to examine and compare resulting performance and robustness characteristics. Performance is gauged by the amount of residual tilt and tip present on the wavefront after exiting the controlled system, while robustness to various levels of atmospheric turbulence, control loop processing time delays, sensor noise, and sensor saturation is also evaluated. The controller design is limited to the use of linear models even though components within the control loop have some nonlinear characteristics; however, consideration has been given to the nonlinear aspects of the system via the simulation environment in order to observe the linear controller in a near-real-world environment. The optimal control algorithm is developed in MATLAB[®] software and evaluated within the modeled control loop using Simulink[®]. The average angular residual tilt and tip (and associated standard deviation) resulting from the controller design is 15.21 ± 10.96 *nanoradians* and 20.78 ± 15.29 *nanoradians*, respectively.

Acknowledgements

First and foremost, I want to thank my wife and colleague for your love and support as well as professional help. Additionally, I'd like to thank my advisor, Lt Col Juan Vasquez, and thesis committee members, Dr. Peter Maybeck, Lt Col Matthew Goda, and Dr. Jim Brown for the professional guidance and sincere encouragement you provided me. Finally, I offer gratitude to my friends and colleagues, Capt Maynard Porter and Lt Steve Mawhorter, for your numerous fresh problem-solving ideas.

Neil D. Paris

LQG/LTR Tilt and Tip Control for the Starfire Optical Range 3.5-meter Telescope's Adaptive Optics System

I. Introduction

The Air Force Research Laboratory has sponsored research on the tracking control loop portion of the adaptive optics system in the Starfire Optical Range 3.5-meter telescope at Kirtland Air Force Base. The control loop includes two steering mirrors (Coarse Steering Mirror and Fine Steering Mirror) used to remove tilt and tip wavefront phase distortion, caused by the earth's atmosphere, from light entering the telescope. The current control system consists of a separate controller for each of the two mirrors. The control loops for each mirror have been designed independently and operate in parallel without any knowledge of what the other controller and mirror are doing. Since the two mirrors are located in series within the optical train, yet the controllers are completely separate, there is a concern for stability of the two control loops when operating simultaneously. Currently, the Coarse Steering Mirror is controlled by a Linear Quadratic Gaussian controller while the Fine Steering Mirror controller is based on classical control techniques such as pole-placement (5). A thorough study of a single controller design to control both mirrors with guaranteed stability and evaluated performance has not been conducted. The goal of this thesis is to design an optimal controller with some type of guaranteed stability to control both mirrors simultaneously and to evaluate the controller's robustness and performance.

1.1 Problem Statement

The objective of this research is to design a single Linear Quadratic Gaussian controller to control both steering mirrors and to evaluate the stability robustness and performance of the controller through simulation and Monte Carlo analysis. Controller design elements and simulation parameters will be varied to examine and compare resulting performance and robustness characteristics.

1.2 *Scope of the Research*

The controller design is limited to the use of linear models even though components within the control loop have some nonlinear characteristics. However, consideration has been given to the nonlinear aspects of the system via the simulation environment in order to observe the linear controller in a near-real-world environment. The controller is designed such that the control system attempts to eliminate wavefront tilt and tip induced by the earth's atmosphere. Other sources of tilt and tip, such as mount jitter and off-center viewing, are not considered to narrow the scope of the research. Similarly, the simulations only produce tilt and tip wavefront values caused by the atmosphere. The significance of neglecting other sources of tilt and tip is not known; however, future consideration of these sources is recommended in Chapter 5. The following are specific simplifications and assumptions made in this research:

- It is assumed that the dynamics of the steering mirrors do not change based on the size of the received inputs, although in reality the dynamics are nonlinear in that they differ with small versus large inputs (6). Data supporting the nonlinear characteristics of the mirrors is not available for this research; therefore, the dynamics corresponding to small inputs are used for both the controller models and to represent the mirrors in simulation. Section 4.1.2.3 discusses the way in which simulations are conducted to validate the use of the small input dynamics.
- To reduce simulation time and allow an adequate amount of data collection for proper analysis, a reduced order truth model of the steering mirrors is used in simulation versus a higher order, more accurate representation of the actual mirror dynamics. The effect this has on the fidelity of the results is unclear; however, further recommendations are provided in Chapter 5.
- It is assumed that the pyramid sensor, which provides measurements of wavefront tilt and tip to the controller, is corrupted by white Gaussian noise.
- The nonlinear region of the sensor is represented in simulation as a single saturation value outside of the linear region.

- Controller performance using time a time-varying Kalman filter and associated control input gains is not simulated; only constant steady-state gains are used.
- The processing time delays resulting from various components of the control system are represented in simulation as a multiple sample delay at one location in the control loop. The provided tilt and tip measurements to the controller are delayed to simulate pyramid sensor and control algorithm processing times, and to test the latency limits of the control system.
- The Air Force Research Laboratory provided a single dynamics model to represent both axes of the Coarse Steering Mirror. Experimental data to develop independent models were not available until the late stages of research, and Section 3.3.2.1 portrays the similarity of the that data in each axis, justifying the use of a single model.
- Each steering mirror induces tilt and tip on the wavefront equal to twice the deflection angles of the mirror (6); however, it is simplified such that the induced tilt and tip are equal to the deflection angles.
- It is assumed that the adaptive optics system has removed all aberrations from the wavefront with the exception of tilt and tip so that beam spreading does not occur at the sensor.
- The wavefront phase distortion occurring as a result of the light passing through the entire atmosphere to the aperture is represented by a single phase screen at the aperture. The effects of amplitude variation of the light is ignored when using only one phase screen; however, amplitude variation has negligible impact on tilt and tip for average atmospheric turbulence simulated in this research. An average atmospheric wind speed is used to propagate the phase screen, which does not represent altitude-dependent evolution of the turbulence; however, evolution of the turbulence over the short simulation times of one and four seconds used in this research, has negligible impact on tilt and tip.
- The telescope is pointed at zenith since the simulated phase screen represents a vertical propagation path (normal to the aperture) through the atmosphere.

1.3 Research Contributions

The analysis of the research results provides a comparison of the performance and robustness properties of the LQG/LTR controller design with various levels of control authority given to the Coarse and Fine Steering Mirrors. The determined levels of performance and robustness portray the ability of the control system to remove tilt and tip phase distortion from collected light. The actual tracking system components of the Starfire Optical Range 3.5-meter Telescope are modeled within the LQG/LTR controller design, as well as in simulation; therefore, implementation of the controller design into the telescope is possible. The analysis from this research can be used by the Air Force Research Laboratory to compare with current tracking performance and stability.

1.4 Thesis Overview

Chapter 2 provides a fundamental overview of atmospheric turbulence (as it relates to optics), adaptive optics, and optimal control. Atmospheric turbulence as it relates to the wavefront of propagating light is explained with emphasis placed on defining what is meant by the tilt and tip of the wavefront. The primary components of the adaptive optics system of the 3.5-meter telescope are described, and the characteristics of the elements of the steering mirror control system are presented. Lastly, the definition of optimal control along with a brief description of several design techniques is presented, and the design technique used in this research is identified.

Chapter 3 presents the details of the controller design preceded by the theory upon which the research and design are based. Optical theory and control theory pertaining to the control design and the simulations developed are explained. In particular, theory behind atmospheric turbulence is the basis of a simulation method used to develop the tilt and tip truth data. The remaining theory supports the development of the controller design and the simulation used to evaluate the controller's performance. The details of the controller design include the specific equations and algorithms of the control loop.

Chapter 4 describes the simulations performed and presents the results along with analysis. One simulation is used to develop the actual tilt and tip of the wavefront that the

control system is designed to eliminate. The primary simulation incorporates the steering mirrors, controller, sensor, and actual tilt and tip values into a control loop to determine the performance of the controller. Noise and saturation of the sensor, as well as time delays to represent online processing time, are included in the control loop simulation. Controller design elements and simulation parameters are varied to examine and compare resulting performance and robustness characteristics. The final results are developed from a Monte Carlo analysis in which the performance of the controller is evaluated over multiple simulation runs.

Chapter 5 contains stated conclusions and recommendations based on the results presented in Chapter 4. Included is a summary of the research and the results, followed by conclusions drawn from analysis of the results. For future work, the recommendations address areas of the controller design and simulation in this research that could be made to have higher fidelity and/or more robustness.

II. Background

2.1 Atmospheric Phase Distortion

The Earth’s atmosphere is a mixture of various gaseous elements and thus has an index of refraction that is not uniform. Temperature, pressure, air density, and wind all contribute to vary the index of refraction spatially throughout the atmosphere in a stochastic manner. Index of refraction is the ratio of the speed of light in a vacuum to the speed of light in a given medium. Since the index of refraction is not uniform throughout the atmosphere, the speed of light varies throughout the atmosphere. A light beam traveling through the atmosphere can be thought of as a collection of many light rays. As light rays that eventually reach a ground-based telescope aperture pass through parts of the atmosphere that differ in index of refraction, the rays are delayed with respect to each other such that they reach the aperture at different times (since they travel at varying speeds) as shown in Figure 2.1. The time delays correspond to phase differences (thus phase distortion) across the wavefront of the light beam that cause the final image resulting from an optical system such as a telescope to be distorted. The phase distortion property of the atmosphere is generally referred to as turbulence. A distorted wavefront can also be thought of as a sheet with wrinkles as shown in Figure 2.2, which is the two-dimensional version of the distorted wavefront in Figure 2.1.

The overall phase distortion of a wavefront can be decomposed into different modes of phase aberrations through Zernike decomposition (14). The first mode of the Zernike decomposition, piston, is the aperture-averaged wavefront phase, to which the telescope system is insensitive; therefore piston will not be considered. The second and third modes of the Zernike decomposition are referred to as tilt and tip. Tilt represents the average deflection in phase of the distorted wavefront from the horizontal axis of a non-distorted wavefront, and tip represents the average deflection from the vertical axis. The higher order modes increase in complexity. Most of the wavefront phase distortion caused by the atmosphere is in the tilt and tip modes. Atmosphere-induced tilt and tip causes the image to “dance” around the center of a camera’s imaging plane. Images are often recorded over long exposure time intervals, during which the instantaneous images move around,

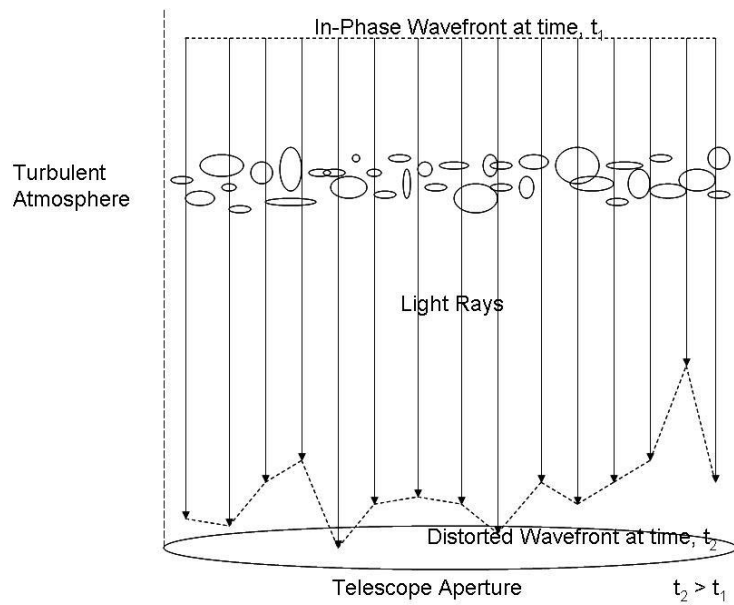


Figure 2.1 Effect of Atmospheric Turbulence: At time t_1 , all of the wavefront has the identical phase. At time t_2 , after passing through the atmosphere, the phase is no longer identical across all of the wavefront.

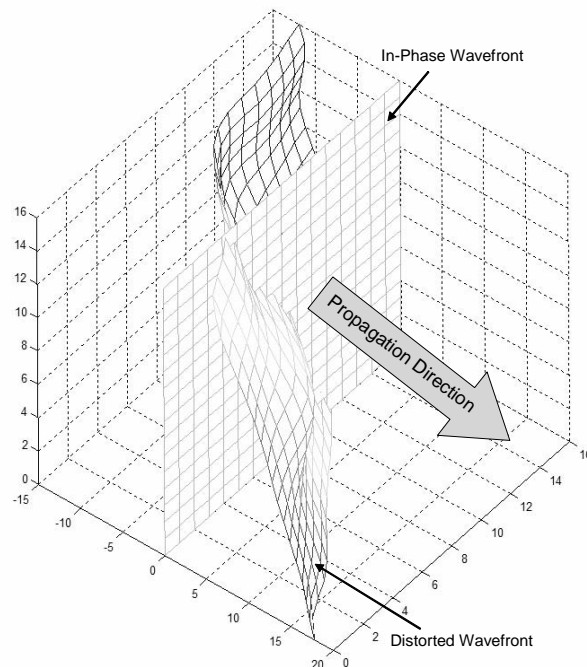


Figure 2.2 Non-disturbed vs. Disturbed Wavefront: The in-phase wavefront depicts what the distorted wavefront would look like if it had not been disturbed by the atmosphere.

causing the resulting long exposure image to be blurry. Removing the tilt and tip from the wavefront before reaching the camera keeps the instantaneous images in the same location, thus eliminating the blurring effect in the long exposure image. About 64 percent of the root-mean-squared phase error across any aperture comes from the tilt and tip modes (4, 14, 17).

2.2 Adaptive Optics

2.2.1 General Description. An adaptive optics system typically uses the technique of phase conjugation to remove wavefront aberrations from the incoming light beam (17). An optical beam can be represented by its amplitude, A , and phase, ϕ , and described mathematically by the electric field, $Ae^{-j\phi}$. The distorted phase could be represented as $\phi_{ideal} + \phi_{disturb}$ where ϕ_{ideal} is the proper phase of the field before distortion occurs and $\phi_{disturb}$ is the change in phase due to the distortion. Thus, the phase-distorted field is $Ae^{-j(\phi_{ideal} + \phi_{disturb})}$. The adaptive optics system measures the amount of phase distortion present in the field and applies the conjugate, $Ae^{j\phi_{disturb}}$, to correct for the phase distortion, resulting in the original field, $Ae^{-j\phi_{ideal}}$. An adaptive optics system usually contains the following principle elements: a wavefront sensor(s), a deformable mirror and its control system, and a steering mirror(s) and its control system. A telescope is normally used to collect the incoming light and additional optical elements and cameras are used to record the image. A wavefront sensor measures the phase aberrations on the wavefront of an incoming beam to be used by the deformable mirror and steering mirror control systems. The deformable mirror is typically used to provide compensation for the higher-order phase aberrations (all aberrations except for tilt and tip) by physically forming into the conjugate of the aberrations. The steering mirror removes the average tilt and tip across the beam's wavefront as measured by a wavefront sensor. A deformable mirror contains numerous actuators that push and pull the underside of its surface to form the mirror into the desired shape. A steering mirror is flat and does not alter its surface; rather, it is rotated and tipped to the desired position. Both the deformable mirror and steering mirror control systems are closed-loop; therefore, the time span from measurement to determination

of the inputs to the mirrors must be much smaller than the time it takes the wavefront aberrations to change (4, 17).

2.2.2 Starfire Optical Range 3.5-meter Telescope. The Air Force Research Laboratory (AFRL) operates a 3.5-meter telescope at the Starfire Optical Range (SOR) on Kirtland Air Force Base (see Figure 2.3). The 3.5-meter telescope is equipped with an adaptive optics system designed for satellite tracking and obtaining high-resolution images at visible wavelengths (2). Light first passed through the telescope's adaptive optics system in September of 1997. The telescope is an open-framed structure with a 3.5-meter diameter parabolic primary mirror and a hyperbolic secondary mirror. The telescope is enclosed in a dome that collapses to expose the telescope to the outside environment. Light is passed

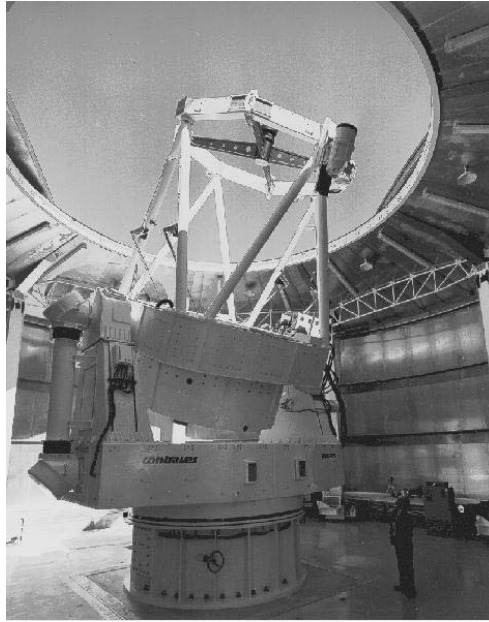


Figure 2.3 SOR 3.5-meter Telescope (1)

through the telescope and reflected down through a window into the coudé rooms below the telescope mount as shown in Figure 2.4. The coudé rooms contain the toroidal optical relay used to interface the light beam with the components of the adaptive optics system, such as the 941-actuator deformable mirror, steering mirrors, and wavefront sensors. After reflection off the K-Mirror, T1 relays the light beam to the Fine Steering Mirror. From the Fine Steering Mirror, the light is relayed to the Coarse Steering Mirror. Mirrors T2

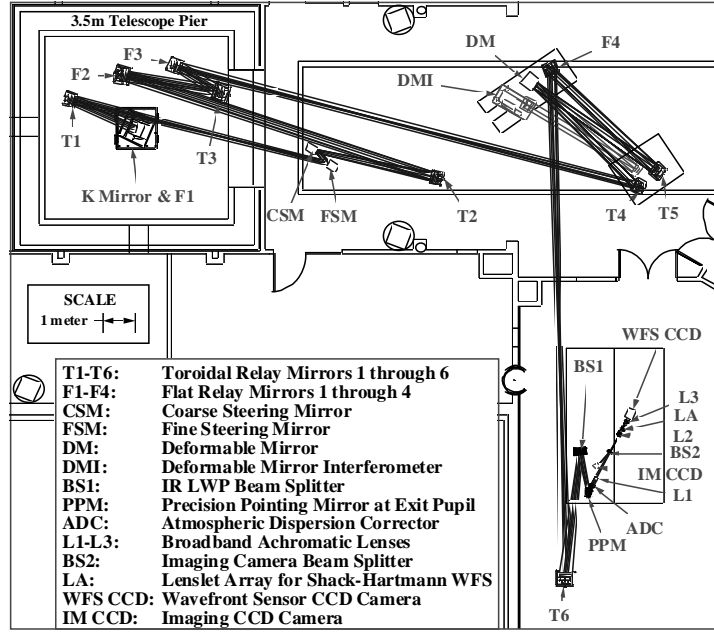


Figure 2.4 Optical Relay Path (7)

through T4 relay the beam to the deformable mirror (DM), followed by T5 and T6 which send the light to the sensor optical bench that includes the wavefront sensor (WFS CCD) and imaging camera (IM CCD) (9, 16).

Within the 3.5-meter telescope's adaptive optics system, the steering mirrors and deformable mirror are actively controlled to compensate for tilt/tip and higher-order aberrations respectively using measurements of the closed-loop errors from the wavefront sensors (15). A Shack-Hartmann Wavefront Sensor (WFS) provides the needed wavefront measurements to control the deformable mirror. The WFS consists of 30 lenslets across the pupil diameter that form a spot array that is relayed to a Charged Coupled Device (CCD) camera. Each lenslet coincides with a 4-by-4 pixel section of the CCD camera, of which the central 2-by-2 region serves as a quad cell for the local tilt/tip calculation. The tilt/tip calculations across all of the lenslets are reconstructed into a phase estimate used to command the deformable mirror actuators to form the mirror's surface into the conjugate of the incoming wavefront to remove the higher-order aberrations in the wavefront. Two steering mirrors serve to remove the tilt/tip aberrations from the wavefront, and the tilt/tip control loop (also referred to as the tracking loop) is the subject of this

research. It is important to note that the tracking loop is separate from the deformable mirror control loop. The Fine Steering Mirror (FSM) has limited angular dynamic range, but a very quick temporal response. The Coarse Steering Mirror (CSM) has a slower response time, but a larger dynamic range. Clearly, some combination of these two mirrors can be useful to eliminate the time-varying wavefront tilt and tip. The WFS can be used to supply tilt/tip correction measurements for the tracking loop by averaging the tilt/tip gradient data across all of the lenslets. However, the WFS is typically not used due to its lower sampling rate of 1500 Hz. Instead, a pyramid sensor is used as the primary tracking sensor. For the purpose of this research, the pyramid sensor is the only sensor considered in the design of the tracking loop. The optical pyramid sensor creates a quad cell effect by splitting the light beam into four directions leading to four photon-counting avalanche photo-diodes. The measurements from the photo-diodes are used in a standard quad cell algorithm to determine the tilt and tip of the wavefront used by the tracking loop. The sampling rate of the pyramid sensor is 5 kHz, therefore, the entire tracking loop operates at that rate (5, 9).

2.2.2.1 CSM. As previously mentioned, the CSM has high dynamic range and low bandwidth; therefore, the CSM compensates for low-frequency tilt/tip disturbances. The CSM exhibits a very lightly damped response to a step input. To correct this undesirable response, a cage loop was designed around the CSM. The cage loop was designed by Dr. James Brown at the SOR and is made up of pure integrators augmented to the mirror's inputs and a Linear Quadratic Gaussian / Loop Transfer Recovery controller. The CSM and its cage loop together are considered as the CSM plant for this research and are described in more detail in the next chapter (5).

2.2.2.2 FSM. The FSM has low dynamic range and high bandwidth; therefore, the FSM compensates for higher-frequency tilt/tip disturbances. The FSM has built-in electronics that act as a cage loop to condition the dynamics of the mirror. The FSM and its built-in electronics together are considered as the FSM plant for this research. The magnitude of the FSM plant's frequency response consists of several steep rises and falls at the higher frequencies. The peaks indicate that the plant has high-

frequency oscillations, and the dips are the result of the electronics attempting to damp the oscillations. Depending on the fit to the experimental frequency response, the resulting transfer function used in linear modeling may in fact represent an unstable plant. Even if the transfer function represents a stable plant, the zeros are in the right-half plane and the poles are very close to the imaginary axis and could easily be pushed into the right-half plane by an improperly designed control system causing the system to be unstable. The characteristics of the FSM are discussed further in the next chapter (5).

2.3 Optimal Control Techniques

2.3.1 Optimal Control Description. In a basic sense, optimal control requires the development of a control law for a particular plant that provides the best possible performance. In order to know when the best possible performance has been attained, some type of measure of performance must be developed. Often the measure of performance comes from a cost function. A cost function can be dependent on the plant, test conditions (such as initial conditions and disturbance inputs), and the control law. The plant and test conditions do not change for a given problem; therefore, the control law can be iteratively designed with the goal of minimizing a particular cost function. The optimal control law is the one that minimizes the cost function. It is the optimal control with respect to that particular cost function; therefore, it is very important to develop a cost function that accurately reflects the overall design requirements (8).

2.3.2 Linear Quadratic Gaussian Control with Loop Transfer Recovery. The Linear Quadratic Gaussian (LQG) control design technique provides an optimal solution to the control problem consisting of a linear plant and measuring device model, a quadratic cost function, random initial conditions, white Gaussian noise inputs, and noisy measurements. The use of a quadratic cost function provides for the ability to find the optimal control law with a reasonable amount of effort from the designer. The basic LQG controller consists of a Kalman filter and a Linear Quadratic Regulator (LQR) in cascade. What makes LQG design so attractive is that it follows the separation principle: the LQR and Kalman filter can each be designed separately, entirely independent of each other.

The LQR control design technique provides an optimal solution to the control problem consisting of a linear plant model, a quadratic cost function, initial conditions, and measurements of the entire state (measured perfectly). The optimal control is found through the minimization of the cost function, which yields the optimal feedback gain matrix. The state vector, \mathbf{x} , is multiplied by the negative of the feedback gain, \mathbf{G}_c , to give the optimal control inputs, \mathbf{u} , as (8)

$$\mathbf{u} = -\mathbf{G}_c \mathbf{x}$$

LQR guarantees a minimum degree of stability robustness (assuming perfect access to the entire state). The concepts of LQR design can easily be extended to the Stochastic Regulator (SR) which accounts for random initial conditions and random white noise disturbance inputs. The cost function for the SR differs from that of the LQR due to the stochastic components involved; however, the SR's cost function remains quadratic. It turns out that the optimal feedback gain matrix of the LQR also minimizes the SR's cost function.

Given a noisy measurement(s) and any known inputs to the plant, a Kalman filter provides estimates of the states, $\hat{\mathbf{x}}$, for the LQR when the states cannot be measured perfectly and entirely, which is often the case. The state estimates from the Kalman filter are optimal in that the mean square estimation error has been minimized.

So within the LQG design, the LQR (in the form of an SR) provides the optimal control law, based on the cost function. The optimal feedback gain matrix is attained from minimization of the cost function. The Kalman filter provides optimal estimates of the states, which are used with the optimal gain matrix in the LQR control law to determine the optimal control inputs given by

$$\mathbf{u} = -\mathbf{G}_c \hat{\mathbf{x}}$$

Therefore, LQG design can be used for many control applications with the major limitations being that the plant model must be linear and driven by white Gaussian noise. LQG design is typically very robust, but it is not guaranteed to have good robustness as when designing an LQR alone. To retain much of the robustness from the LQR, the LQG con-

troller can be modified using Loop Transfer Recovery (LTR). The LTR method is meant to asymptotically recover the robustness to input-multiplicative perturbations (in terms of the stability margins) normally guaranteed by the LQR. Once LTR is implemented, the controller becomes suboptimal at nominal design conditions due to the modification of the original LQG design (the cost is no longer completely optimized). Thus the LTR design results in a compromise between performance (optimal performance was gained by the LQG design) and robustness (8).

The LQG/LTR theory just presented is meant to be an overview and will be developed in greater detail in the next chapter.

2.3.3 H_∞ . A counterpart to LQG design is the H_∞ controller. The H_∞ control design technique uses partial state measurements corrupted by noise to provide the optimal or sub-optimal control with respect to the system infinity norm as a cost function (8). Like LQG, the H_∞ design generates a controller based on the feedback of all states and an estimator. However, the H_∞ feedback controller is referred to as a full information controller since the disturbance input, which does not have to be white noise, is included in the design model. Furthermore, by adding the disturbance to the design model, H_∞ controller dimension can increase significantly above that of the original plant. The H_∞ estimator is fundamentally different from the Kalman filter in two ways (8). First, the H_∞ estimator minimizes the ∞ -norm (cost function) of the gain between a set of input disturbances and the estimation error. Therefore, the H_∞ estimator minimizes the worst-case gain, whereas the Kalman filter minimizes the mean square estimation error. Secondly, the Kalman filter can estimate any linear combination of the states from the individual state estimates since minimum mean square error estimation possesses the property of linear operations; however, minimal ∞ -norm estimation does not possess this property, and it estimates the output of the plant. As a result of the latter remark, the H_∞ estimator gain depends on the linear combination of the states being estimated. Therefore, within the overall H_∞ output feedback controller design, it is more appropriate to estimate the desired control input. The H_∞ controller structure does not benefit from the separation principle since the estimator is dependent on the full information controller design. Analogous to the

use of LTR in LQG design, different types of perturbations (such as input-multiplicative) can be added to the H_∞ controller design to increase robustness. However, the resulting design may still be sensitive to unmodeled perturbations unlike LQG/LTR which has guaranteed asymptotic robustness properties (8).

2.3.4 Implemented Controller Design Technique. In this research, the LQG/LTR design methodology is implemented due to the ease of generating optimal solutions from the quadratic cost function and the robustness that may to be attained through the LTR methodology. Often H_∞ control is used when the system specifications are in terms of bounds on the output, which does not apply to the research being conducted. H_∞ control can be used as an alternative if the LQG control is not able to achieve all of the desired properties; however, the H_∞ controller is not guaranteed to achieve results superior to the LQG design (8).

III. Theory and Controller Design

3.1 Atmospheric Turbulence Theory

This section presents the detailed theory of atmospheric turbulence that primarily serves as the foundation for the simulation used to develop the wavefront tilt and tip values. The theory is also used to develop particular aspects of the Kalman filter model within the final LQG controller design.

3.1.1 Atmospheric Turbulence Strength. As mentioned in Chapter 2, atmospheric turbulence is a stochastic process due to the random fluctuations in the index of refraction. The process is both spatially (three dimensionally) and temporally stochastic. The structure function is a spatial statistic of the atmospheric turbulence and is defined by (4)

$$D_\phi(\rho) = E [(\phi(\rho_1) - \phi(\rho_1 - \rho))^2] \quad (3.1)$$

where ϕ is the phase of light propagating through the atmosphere, ρ_1 is a spatial position and ρ is a spatial distance. Atmospheric turbulence is statistically homogenous and isotropic (spatially), meaning that the moments are invariant to translation and rotation, making D_ϕ as a function of only the scalar ρ valid. The structure function can also be written as (14)

$$D_\phi(\rho) = 2.91k^2\rho^{5/3} \int_{h_{\min}}^{h_{\max}} C_n^2(h)dh \quad (3.2)$$

where $k = \frac{\lambda}{2\pi}$ (λ is the wavelength of the light), h is altitude, and $C_n^2(h)$ is the index of refraction structure constant. In this research, $C_n^2(h)$ is defined by the Hufnagel-Valley (H-V) model (4):

$$C_n^2(h) = 0.00594\left(\frac{v}{27}\right)^2(10^{-5}h)^{10}e^{-h/1000} + 2.7(10^{-16})e^{-h/1500} + 1.7(10^{-14})e^{-h/100} \quad (3.3)$$

where v is the average wind speed throughout the atmosphere. The value of $C_n^2(h)$ is essentially zero when $h > 30km$. Another parameter used to define the strength of atmospheric turbulence is the Fried radius r_0 , defined as the distance ρ at which $D_\phi(\rho) = 2(2.1^{5/3})$

which results in

$$r_0 = 2.1 \left[1.46k^2 \int_{h_{\min}}^{h_{\max}} C_n^2(h) dh \right]^{-3/5} \quad (3.4)$$

The Fried radius represents the aperture radius beyond which a larger aperture size provides no improvement in resolution (14). The Fried radius decreases with stronger turbulence. Atmospheric turbulence can also be defined by the von Karman Power Spectral Density (PSD) Φ_n in terms of the spatial frequency κ (4):

$$\Phi_n(\kappa) = 0.4916693r_0^{-5/3} \frac{e^{-(\kappa/\kappa_m)^2}}{(\kappa^2 + \kappa_0^2)^{11/6}} \quad (3.5)$$

where $\kappa_m = \frac{5.92}{l_0}$ and $\kappa_0 = \frac{1}{L_0}$. The constants, L_0 and l_0 , are called the outer scale and inner scale respectively. The inner scale is the size of a viscous fluid or "eddy", which represents atmospheric air mass motion, at which the fluid transitions from turbulent to stable flow. The outer scale is the eddy size above which turbulent energy is injected. The inner scale decreases with stronger turbulence, while the outer scale increases with stronger turbulence. A detailed discussion of the inner and outer scales is beyond the scope of this research, but can be found in references (4) and (14).

3.1.2 Zernike Polynomials. Chapter 2 mentions that phase aberrations caused by turbulent atmosphere can be decomposed into different modes including tilt and tip. Phase values, $\phi(r, \theta)$, on the wavefront across the entire aperture can be represented by the sum of Zernike polynomials, $Z_i(r, \theta)$, as (14)

$$\phi(r, \theta) = \sum_{i=1}^{\infty} a_i Z_i(r, \theta) \quad (3.6)$$

where (r, θ) are polar coordinates across the circular aperture with $r \leq 1$, i is the individual mode and a_i is the Zernike coefficient. The phase values of the individual modes are defined by $a_i Z_i(r, \theta)$. As mentioned earlier, tilt and tip are modes 2 and 3, respectively, thus the phase contribution from each is

$$\phi_{tilt}(r, \theta) = a_2 Z_2(r, \theta) \quad (3.7)$$

$$\phi_{tip}(r, \theta) = a_3 Z_3(r, \theta) \quad (3.8)$$

It is important to note that the Zernike polynomials are a set of orthonormal basis functions. The Zernike coefficient a_i is a normally distributed random variable of mean zero and variance unique to each mode. The Zernike coefficients for tilt and tip both have a variance defined by

$$E[a_{tilt/tip}^2] = 0.448 \left(\frac{d}{r_0}\right)^{5/3} rad^2 \quad (3.9)$$

where d is the diameter of the aperture and r_0 is the Fried radius defined by Equation (3.4). The tilt and tip coefficients are independent of each other, therefore $E[a_2 a_3] = 0$. The tilt and tip Zernike polynomials are defined by

$$Z_2(r, \theta) = 2r \cos \theta \quad (3.10)$$

$$Z_3(r, \theta) = 2r \sin \theta \quad (3.11)$$

The phase contribution $\phi_{tilt/tip}(r, \theta)$ from tilt or tip is centered about the center of the aperture such that the phase values are negatively symmetric. Zernike decomposition is useful in that the phase contribution from a single mode can be extracted from the total phase resulting from the sum of all contributing modes. Since the Zernike polynomials are orthonormal, the coefficients can be found by projections

$$a_i = \int W(\alpha) \phi\left(\frac{d}{2}\alpha, \theta\right) Z_i(\alpha, \theta) d\alpha \quad (3.12)$$

where $\alpha = \frac{r}{d/2}$ and $W(\alpha)$ is an aperture weighting function normalized such that $\int W(\alpha) d\alpha = 1$. The use of α enables Equation (3.12) to be used with any size aperture. Therefore, the contribution from tilt or tip can be extracted from an aberrated wavefront resulting from many modes of aberration.

3.1.3 Greenwood Frequency. A parameter that provides an indication of how quickly the turbulence changes over time is called the Greenwood frequency, f_G , described by (17)

$$f_G = 2.31 \lambda^{-6/5} \left[\int_{h_{\min}}^{h_{\max}} C_n^2(h) v^{5/3}(h) dh \right]^{3/5}$$

where $v(h)$ is the wind speed as a function of altitude, and h_{\max} and h_{\min} are the maximum and minimum altitudes, respectively, of the atmospheric region of interest. In this research, an average wind speed over all altitudes is used (as stated in Section 1.2); therefore, the equation simplifies to

$$f_G = 0.43\left(\frac{v_{avg}}{r_0}\right) \quad (3.13)$$

using the definition of r_0 in Equation (3.4).

3.2 LQG Theory

This section presents the detailed theory of LQG control synthesis that serves as the foundation for the specific controller designed in this research. In the following subsections, the theory is presented for each component of an LQG controller.

3.2.1 LQR. As mentioned earlier, the LQR controller provides an optimal solution to the deterministic control problem consisting of a linear plant model, a quadratic cost function, initial conditions, and measurements of the entire state. The dynamics of the controlled plant are modeled by a linear dynamics differential equation of the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

where $\mathbf{x}(t)$ is the state vector, $\mathbf{u}(t)$ is the input vector representing inputs to the plant, \mathbf{A} is the plant matrix, and \mathbf{B} is the input matrix. The output of the plant is modeled by the output equation

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$$

where $\mathbf{y}(t)$ is the output vector and \mathbf{C} is the output matrix. The measurements provided to the LQR are modeled by the measurement equation

$$\mathbf{z}(t) = \mathbf{H}\mathbf{x}(t)$$

where $\mathbf{z}(t)$ is the measurement vector and \mathbf{H} is the measurement matrix. In the case of the LQR, $\mathbf{H} = \mathbf{C} = \mathbf{I}$ (where \mathbf{I} is the identity matrix) since the measurements are of

the entire state. The matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{H} are all constant-valued since attention will be restricted to time-invariant systems only, though LQG control is not restricted to time-invariant systems. The model equations just presented are state-space equations and can be developed from the transfer function of the plant. The plant dynamics equation is propagated forward from the initial condition $\mathbf{x}(0) = \mathbf{x}_0$. The quadratic cost function to be minimized is (8)

$$J(\mathbf{x}(t), \mathbf{u}(t)) = \frac{1}{2} \mathbf{x}^T(t_f) \mathbf{X}_f \mathbf{x}(t_f) + \frac{1}{2} \int_0^{t_f} (\mathbf{x}^T(t) \mathbf{W}_{xx} \mathbf{x}(t) + \mathbf{u}^T(t) \mathbf{W}_{uu} \mathbf{u}(t)) dt \quad (3.14)$$

subject to the initial condition $\mathbf{x}(0) = \mathbf{x}_0$. The matrix \mathbf{W}_{xx} is the state weighting matrix, \mathbf{W}_{uu} is the control weighting matrix, and \mathbf{X}_f is the state weighting matrix at the final time. Both \mathbf{W}_{xx} and \mathbf{X}_f are positive semidefinite and \mathbf{W}_{uu} is positive definite. The weighting matrices can be diagonal matrices that determine the importance of driving each state to zero (in the case of \mathbf{W}_{xx} and \mathbf{X}_f) and the level of control input (in the case of \mathbf{W}_{uu}). The first term in Equation (3.14) represents the cost or penalty obtained by failing to drive the state vector to zero at the final time, t_f . The second term represents the penalty accumulated over time from zero to t_f obtained by failing to drive the state vector to zero during this time interval and the penalty obtained by exerting control during the same interval. The optimal control to apply to the plant at time t is the $\mathbf{u}(t)$ vector that minimizes the cost function. This is a constrained optimal control problem, with the dynamics state equation providing a family of constraint equations. The optimal control is unique to a particular cost function such that if \mathbf{W}_{xx} , \mathbf{X}_f , and/or \mathbf{W}_{uu} are altered, then the optimal control in general will change. Therefore, it is important to determine the best cost function to use for a given control problem.

One technique to develop the expression for the optimal control is through the use of the Hamiltonian system. The optimal control problem just described can be converted to an unconstrained problem through the use of Lagrange multipliers to augment the cost function (8). Solving for the optimal control input from the augmented cost function and applying it to the original plant state-space dynamics model results in an augmented state-space system called the Hamiltonian system. The Hamiltonian equations provide a

set of necessary and sufficient conditions to determine the control that minimizes the cost function. The optimal control is (8)

$$\mathbf{u}(t) = -\mathbf{W}_{uu}^{-1}\mathbf{B}^T\mathbf{S}(t)\mathbf{x}(t) = -\mathbf{G}_c(t)\mathbf{x}(t)$$

where $\mathbf{S}(t)$, called the matrix of proportionality, is completely determined from the state transition matrix of the Hamiltonian system (8). The optimal feedback gain matrix, $\mathbf{G}_c(t)$, is given by

$$\mathbf{G}_c(t) = \mathbf{W}_{uu}^{-1}\mathbf{B}^T\mathbf{S}(t)$$

Note that $\mathbf{S}(t)$ is the matrix of proportionality between the costate and the state. The costate can be found by solving the homogeneous Hamiltonian state equation which yields $\mathbf{s}(t) = \mathbf{S}(t)\mathbf{x}(t)$ where $\mathbf{s}(t)$ is the costate. As mentioned earlier, $\mathbf{S}(t)$ is determined by the Hamiltonian state transition matrix. However, determining this state transition matrix is typically very difficult. Therefore, $\mathbf{S}(t)$ can alternatively be found by solving the nonlinear Riccati differential equation generated by taking the derivative of $\mathbf{s}(t)$

$$\dot{\mathbf{S}}(t) = -\mathbf{S}(t)\mathbf{A} - \mathbf{A}^T\mathbf{S}(t) - \mathbf{W}_{xx} + \mathbf{S}(t)\mathbf{B}\mathbf{W}_{uu}^{-1}\mathbf{B}^T\mathbf{S}(t) \quad (3.15)$$

backward in time from the final condition, $\mathbf{S}(t_f) = \mathbf{X}_f$.

3.2.1.1 Steady-State LQR. The optimal feedback gains experience transients over a period of time close to the final time, but reach steady-state values as time is propagated backward from the final time. When the LQR controller is operated for a period of time longer than the transient time, then the steady-state gain values are often adequate for all time. Depending on how long the operation time is compared to the transient time, overall system performance may be degraded when using the steady-state gains. The designer must decide if the degree of degradation is acceptable. Using the steady-state gain matrix simplifies the controller implementation, since only one computation of the gain matrix needs to be performed, rather than computing and storing the gains at each sample time. In the steady-state case, the following cost function can be

used (8):

$$\bar{J}(\mathbf{x}(t), \mathbf{u}(t)) = \frac{1}{2} \int_0^\infty (\mathbf{x}^T(t) \mathbf{W}_{xx} \mathbf{x}(t) + \mathbf{u}^T(t) \mathbf{W}_{uu} \mathbf{u}(t)) dt$$

subject to the initial condition $\mathbf{x}(0) = \mathbf{x}_0$. The \mathbf{X}_f term is no longer included since the final state is not important (the final time is never reached, nor is the terminal transient). The steady-state optimal gain matrix is found by minimizing this cost function. The equation for the steady-state optimal control is identical to the time-varying case except that the matrix of proportionality and the gain matrix are no longer time-varying as shown in Equations (3.16) to (3.18):

$$\mathbf{u}(t) = -\mathbf{W}_{uu}^{-1} \mathbf{B}^T \bar{\mathbf{S}} \mathbf{x}(t) = -\bar{\mathbf{G}}_c \mathbf{x}(t) \quad (3.16)$$

$$\bar{\mathbf{G}}_c = \mathbf{W}_{uu}^{-1} \mathbf{B}^T \bar{\mathbf{S}} \quad (3.17)$$

$\bar{\mathbf{S}}$ is the solution to the steady-state version of Equation (3.15) which is

$$\bar{\mathbf{S}} \mathbf{A} + \mathbf{A}^T \bar{\mathbf{S}} - \bar{\mathbf{S}} \mathbf{B} \mathbf{W}_{uu}^{-1} \mathbf{B}^T \bar{\mathbf{S}} + \mathbf{W}_{xx} = \mathbf{0}. \quad (3.18)$$

A time-invariant closed-loop system results from using the steady-state LQR controller. The state equation of the closed-loop system is

$$\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B}(-\bar{\mathbf{G}}_c \mathbf{x}(t)) = (\mathbf{A} - \mathbf{B} \bar{\mathbf{G}}_c) \mathbf{x}(t)$$

Therefore, the closed-loop poles are found by solving the characteristic equation

$$\det(s\mathbf{I} - \mathbf{A} + \mathbf{B} \bar{\mathbf{G}}_c) = 0$$

and these poles provide insight regarding stability and performance.

3.2.1.2 Stochastic Regulator. The Stochastic Regulator (SR) accounts for random initial conditions and random white noise disturbance inputs. Therefore, the plant

dynamics equation is now

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{G}\mathbf{w}(t) \quad (3.19)$$

where $\mathbf{w}(t)$ is a zero-mean white Gaussian dynamics noise vector (input disturbance) of strength \mathbf{Q} :

$$\mathbf{Q}(t) = E[\mathbf{w}(t)\mathbf{w}^T(t + \tau)] = \mathbf{Q}\delta(\tau) \quad (3.20)$$

and \mathbf{G} is the constant-valued plant white Gaussian noise (WGN) matrix. The initial condition vector \mathbf{x}_0 is zero-mean and has the correlation matrix

$$\mathbf{\Gamma}_x(0) = E[\mathbf{x}(0)\mathbf{x}^T(0)]$$

The WGN input can represent a physical input noise or can be incorporated to account for inaccurate modeling when the plant model does not exactly represent the true plant dynamics, or both. Due to the stochastic components, the cost function for the SR is now represented as an expected value expression (8):

$$J_{SR}(\mathbf{x}(t), \mathbf{u}(t)) = \frac{1}{2}E \left[\mathbf{x}^T(t_f)\mathbf{X}_f\mathbf{x}(t_f) + \int_0^{t_f} (\mathbf{x}^T(t)\mathbf{W}_{xx}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{W}_{uu}\mathbf{u}(t))dt \right]$$

The optimal feedback gain matrix of the LQR also minimizes the SR's cost function (8, 11). Similarly, the steady-state SR gain matrix is identical to the steady-state LQR gain matrix with the cost function given by

$$\bar{J}_{SR}(\mathbf{x}(t), \mathbf{u}(t)) = \frac{1}{2}E \left[\lim_{t_f \rightarrow \infty} \frac{1}{t_f} \int_0^{t_f} (\mathbf{x}^T(t)\mathbf{W}_{xx}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{W}_{uu}\mathbf{u}(t))dt \right]$$

where $\frac{1}{t_f}$ is a normalizing constant used to ensure the cost function is finite.

3.2.1.3 Discrete-Time Implementation. The theory presented thus far describes a continuous-time LQR; however, the controller is typically implemented using a sampled-data or discrete-time design. The continuous-time plant model must be discretized

as an equivalent discrete-time system resulting in the following model (12):

$$\mathbf{x}(t_{i+1}) = \mathbf{\Phi}(t_{i+1}, t_i) \mathbf{x}(t_i) + \mathbf{B}_d \mathbf{u}(t_i) + \mathbf{w}_d(t_i) \quad (3.21)$$

$$\mathbf{z}(t_i) = \mathbf{H} \mathbf{x}(t_i)$$

where $\mathbf{H} = \mathbf{C} = \mathbf{I}$, and \mathbf{B}_d and \mathbf{w}_d are defined by

$$\mathbf{B}_d = \int_{t_i}^{t_{i+1}} \mathbf{\Phi}(t_{i+1}, \tau) \mathbf{B} d\tau \quad (3.22)$$

$$E[\mathbf{w}_d(t_i)] = \mathbf{0} \quad (3.23)$$

$$E[\mathbf{w}_d(t_i) \mathbf{w}_d^T(t_i)] = \mathbf{Q}_d = \int_{t_i}^{t_{i+1}} \mathbf{\Phi}(t_{i+1}, \tau) \mathbf{G} \mathbf{Q} \mathbf{G}^T \mathbf{\Phi}^T(t_{i+1}, \tau) d\tau \quad (3.24)$$

$$E[\mathbf{w}_d(t_i) \mathbf{w}_d^T(t_j)] = \mathbf{0}, t_i \neq t_j \quad (3.25)$$

noting again the constant-valued system matrices and assuming a constant noise strength \mathbf{Q} . For a time-invariant system, the integrations need to be performed only once if the sample period is constant. The state transition matrix, $\mathbf{\Phi}(t_{i+1}, t_i)$, for a time-invariant system is defined by $\mathbf{\Phi}(t_{i+1}, t_i) = e^{\mathbf{A}(t_{i+1}-t_i)}$ which is constant for a particular sample period (hence the time argument will be dropped from this point forward). Furthermore, the discrete-time cost function is written as (11)

$$J(\mathbf{x}(t_i), \mathbf{u}(t_i)) = \frac{1}{2} E \left[\mathbf{x}^T(t_{N+1}) \mathbf{X}_f \mathbf{x}(t_{N+1}) + \sum_{i=0}^N (\mathbf{x}^T(t_i) \mathbf{X} \mathbf{x}(t_i) + \mathbf{u}^T(t_i) \mathbf{U} \mathbf{u}(t_i)) \right] \quad (3.26)$$

where \mathbf{X} and \mathbf{U} are given by

$$\mathbf{X} = \int_{t_i}^{t_{i+1}} \mathbf{\Phi}^T(t, t_i) \mathbf{W}_{xx} \mathbf{\Phi}(t, t_i) dt$$

$$\mathbf{U} = \int_{t_i}^{t_{i+1}} [\bar{\mathbf{B}}^T(t, t_i) \mathbf{W}_{xx} \bar{\mathbf{B}}(t, t_i) + \mathbf{W}_{uu}] dt$$

and

$$\bar{\mathbf{B}}(t, t_i) \triangleq \int_{t_i}^t \mathbf{\Phi}(t, \tau) \mathbf{B}(\tau) d\tau$$

The optimal control gains are determined from

$$\mathbf{G}_c(t_i) = [\mathbf{U} + \mathbf{B}_d^T \mathbf{K}_c(t_{i+1}) \mathbf{B}_d]^{-1} [\mathbf{B}_d^T \mathbf{K}_c(t_{i+1}) \Phi] \quad (3.27)$$

where $\mathbf{K}_c(t_{i+1})$ is the solution to the backward Riccati difference equation

$$\mathbf{K}_c(t_i) = \mathbf{X} + \Phi^T \mathbf{K}_c(t_{i+1}) \Phi - [\Phi^T \mathbf{K}_c(t_{i+1}) \mathbf{B}_d] [\mathbf{U} + \mathbf{B}_d^T \mathbf{K}_c(t_{i+1}) \mathbf{B}_d]^{-1} [\mathbf{B}_d^T \mathbf{K}_c(t_{i+1}) \Phi] \quad (3.28)$$

subject to $\mathbf{K}_c(t_{N+1}) = \mathbf{X}_f$, and the optimal control input is

$$\mathbf{u}(t_i) = -\mathbf{G}_c(t_i) \mathbf{x}(t_i)$$

The equations for the steady-state solution are

$$\bar{J}(\mathbf{x}(t_i), \mathbf{u}(t_i)) = \frac{1}{2} E \left[\lim_{N \rightarrow \infty} \frac{1}{t_N} \sum_{i=0}^N (\mathbf{x}^T(t_i) \mathbf{X} \mathbf{x}(t_i) + \mathbf{u}^T(t_i) \mathbf{U} \mathbf{u}(t_i)) \right] \quad (3.29)$$

$$\bar{\mathbf{G}}_c = [\mathbf{U} + \mathbf{B}_d^T \bar{\mathbf{K}}_c \mathbf{B}_d]^{-1} [\mathbf{B}_d^T \bar{\mathbf{K}}_c \Phi] \quad (3.30)$$

$$\mathbf{X} + \Phi^T \bar{\mathbf{K}}_c \Phi - [\Phi^T \bar{\mathbf{K}}_c \mathbf{B}_d] [\mathbf{U} + \mathbf{B}_d^T \bar{\mathbf{K}}_c \mathbf{B}_d]^{-1} [\mathbf{B}_d^T \bar{\mathbf{K}}_c \Phi] - \bar{\mathbf{K}}_c = \mathbf{0} \quad (3.31)$$

$$\mathbf{u}(t_i) = -\bar{\mathbf{G}}_c \mathbf{x}(t_i).$$

The discrete-time state equation of the closed-loop system resulting from the steady-state control law is

$$\mathbf{x}(t_{i+1}) = (\Phi - \mathbf{B}_d \bar{\mathbf{G}}_c) \mathbf{x}(t_i)$$

3.2.2 Kalman Filter. Typically, the states of a system cannot be measured perfectly due to noise from the measurement device, and the measurements are often linear combinations of the states. In these cases, the LQR controller cannot be used without the aid of an observer to estimate the true value of each state for use in the control algorithms (8). A Kalman filter is an observer that estimates the states by minimizing the mean square estimation error, $E[(x(t) - \hat{x}(t))^2]$, where $\hat{x}(t)$ is the state estimate. The mean square estimation errors of a state vector make up the elements of the error covariance

matrix $\mathbf{P}(t)$ (12):

$$\mathbf{P}(t) = E [(\mathbf{x}(t) - \hat{\mathbf{x}}(t))(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T | \mathbf{Z}(t)]$$

where $\mathbf{Z}(t)$ is a vector containing actual measurement values obtained up to the current time, t . The Kalman filter uses the plant dynamics equation, the measurement model equation, the statistics of the uncertainties, and the actual measurement values to compute the optimal state estimates. The following discussion will pertain to the *linear* Kalman filter based on the assumptions of a linear system model driven by additive WGN.

The linear plant dynamics model is described by Equations (3.19) to (3.20). The initial condition vector \mathbf{x}_0 is modeled as a normally distributed random vector with mean $E[\mathbf{x}_0] = \hat{\mathbf{x}}_0$ and covariance $E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] = \mathbf{P}_0$. The linear measurement model is

$$\mathbf{z}(t_i) = \mathbf{H}\mathbf{x}(t_i) + \mathbf{v}(t_i)$$

where $\mathbf{z}(t_i)$ is the measurement vector and $\mathbf{v}(t_i)$ represents the measurement noise assumed to be a zero-mean white Gaussian noise vector of covariance:

$$\mathbf{R}(t_i) = E[\mathbf{v}(t_i)\mathbf{v}^T(t_i)] \quad (3.32)$$

The plant dynamics model can be written in discrete-time via Equations (3.21) to (3.25). For the purpose of this research, the initial condition, dynamics noise, and measurement noise are assumed independent of each other.

The Kalman filter uses the present and all past measurements in generating the state estimates. Therefore, the state estimates are conditioned on the complete measurement history. The objective of the Kalman filter is to propagate the optimal estimate from time t_{i-1}^+ , just after the measurement is incorporated into the estimate at time t_{i-1} , to t_i^- , the time just prior to the next measurement being provided to the filter, and finally, to the time, t_i^+ , just after the next measurement is incorporated. The estimate generation can be broken up into two parts: a time propagation from t_{i-1}^+ to t_i^- , and a measurement update

from t_i^- to t_i^+ . The time propagation difference equations are (12)

$$\hat{\mathbf{x}}(t_i^-) = \Phi \hat{\mathbf{x}}(t_{i-1}^+) + \mathbf{B}_d \mathbf{u}(t_{i-1}) \quad (3.33)$$

$$\mathbf{P}(t_i^-) = \Phi \mathbf{P}(t_{i-1}^+) \Phi^T + \mathbf{Q}_d \quad (3.34)$$

The measurement update equations are

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)[\mathbf{z}(t_i) - \mathbf{H}\hat{\mathbf{x}}(t_i^-)] \quad (3.35)$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}\mathbf{P}(t_i^-) \quad (3.36)$$

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T[\mathbf{H}\mathbf{P}(t_i^-)\mathbf{H}^T + \mathbf{R}]^{-1} \quad (3.37)$$

where $\mathbf{K}(t_i)$ is the Kalman filter gain matrix and the measurement noise covariance \mathbf{R} is assumed to be constant.

3.2.2.1 Steady-State Kalman Filter. If the system model is time-invariant (\mathbf{A} , \mathbf{B} , \mathbf{G} , and \mathbf{H} are constant) and the statistics are stationary (\mathbf{Q} and \mathbf{R} are constant), then \mathbf{P} and \mathbf{K} experience an initial transient period followed by a convergence to steady-state values. Thus, if the system is operated for a time longer than the transient time period, then the steady-state Kalman filter could be used as an approximation for all time assuming any performance degradation during the transient period is deemed acceptable. As in the case of using the steady-state LQR, use of the steady-state Kalman filter simplifies the implementation by reducing computation time. A Riccati difference equation used to solve for the steady-state covariance can be formed by finding $\mathbf{P}(t_{i-1}^+)$ in Equation (3.34) from Equation (3.36) resulting in

$$\mathbf{P}(t_{i-1}^+) = \mathbf{P}(t_{i-1}^-) - \mathbf{K}(t_{i-1})\mathbf{H}\mathbf{P}(t_{i-1}^-) \quad (3.38)$$

Substituting Equation (3.37) with t_i replaced with t_{i-1} into Equation (3.38) for $\mathbf{K}(t_{i-1})$ gives

$$\mathbf{P}(t_{i-1}^+) = \mathbf{P}(t_{i-1}^-) - \mathbf{P}(t_{i-1}^-)\mathbf{H}^T[\mathbf{H}\mathbf{P}(t_{i-1}^-)\mathbf{H}^T + \mathbf{R}]^{-1}\mathbf{H}\mathbf{P}(t_{i-1}^-) \quad (3.39)$$

Substituting Equation (3.39) for $\mathbf{P}(t_{i-1}^+)$ in Equation (3.34) results in

$$\begin{aligned}\mathbf{P}(t_i^-) &= \Phi[\mathbf{P}(t_{i-1}^-) - \mathbf{P}(t_{i-1}^-)\mathbf{H}^T[\mathbf{HP}(t_{i-1}^-)\mathbf{H}^T + \mathbf{R}]^{-1}\mathbf{HP}(t_{i-1}^-)]\Phi^T + \mathbf{Q}_d \\ &= \Phi\mathbf{P}(t_{i-1}^-)\Phi^T - \Phi\mathbf{P}(t_{i-1}^-)\mathbf{H}^T[\mathbf{HP}(t_{i-1}^-)\mathbf{H}^T + \mathbf{R}]^{-1}\mathbf{HP}(t_{i-1}^-)\Phi^T + \mathbf{Q}_d\end{aligned}$$

Performing matrix algebra, ignoring all time arguments for convenience, and setting the equation equal to zero produces the steady-state Riccati difference equation for $\bar{\mathbf{P}}$ (the steady-state value of $\mathbf{P}(t_i^-)$):

$$\Phi\bar{\mathbf{P}}\Phi^T - \Phi\bar{\mathbf{P}}\mathbf{H}^T[\mathbf{HPH}^T + \mathbf{R}]^{-1}\mathbf{HP}\Phi^T + \mathbf{Q}_d - \bar{\mathbf{P}} = \mathbf{0} \quad (3.40)$$

From Equation (3.37), the steady-state Kalman filter gain matrix is

$$\bar{\mathbf{K}} = \bar{\mathbf{P}}\mathbf{H}^T[\mathbf{HPH}^T + \mathbf{R}]^{-1} \quad (3.41)$$

3.2.3 LQG/LTR. As mentioned in Chapter 2, the LQG control design technique provides an optimal solution to the control problem consisting of a linear plant model, a quadratic cost function, random initial conditions, white Gaussian noise inputs, and noisy measurements. The LQG controller consists of a Kalman filter cascaded with an LQR (in the form of an SR), and the two components can be designed independently due to the separation principle (8, 11). The optimal control input provided to the plant is determined by minimizing the cost function described by Equation (3.26) or (3.29) with $\mathbf{x}(t_i)$ replaced with $\hat{\mathbf{x}}(t_i^+)$, the state estimates provided by the Kalman filter, resulting in

$$J_{LQG}(\hat{\mathbf{x}}(t_i^+), \mathbf{u}(t_i)) = \frac{1}{2}E \left[\hat{\mathbf{x}}^T(t_{N+1}^+) \mathbf{X}_f \hat{\mathbf{x}}(t_{N+1}^+) + \sum_{i=0}^N (\hat{\mathbf{x}}^T(t_i^+) \mathbf{X} \hat{\mathbf{x}}(t_i^+) + \mathbf{u}^T(t_i) \mathbf{U} \mathbf{u}(t_i)) \right] \quad (3.42)$$

and

$$\bar{J}_{LQG}(\hat{\mathbf{x}}(t_i^+), \mathbf{u}(t_i)) = \frac{1}{2}E \left[\lim_{N \rightarrow \infty} \frac{1}{t_N} \sum_{i=0}^N (\hat{\mathbf{x}}^T(t_i^+) \mathbf{X} \hat{\mathbf{x}}(t_i^+) + \mathbf{u}^T(t_i) \mathbf{U} \mathbf{u}(t_i)) \right] \quad (3.43)$$

Therefore the optimal control law is

$$\mathbf{u}(t_i) = -\mathbf{G}_c(t_i)\hat{\mathbf{x}}(t_i^+)$$

or

$$\mathbf{u}(t_i) = -\bar{\mathbf{G}}_c\hat{\mathbf{x}}(t_i^+)$$

if the steady-state gains are being used, where $\mathbf{G}_c(t_i)$ is determined by Equations (3.27) and (3.28) and $\bar{\mathbf{G}}_c$ by Equations (3.30) and (3.31).

As mentioned in Chapter 2, LTR is an attempt to recover the robustness normally guaranteed by the LQR, but at the expense of adding to the cost and thus limiting optimal performance of the system at design conditions. One such LTR procedure consists of adding fictitious noise, $\mathbf{w}_f(t)$, to the control input within the plant model used by the Kalman filter, resulting in the new plant model (8)

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \begin{bmatrix} \mathbf{G} & \vdots & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{w}(t) \\ \dots \\ \mathbf{w}_f(t) \end{bmatrix}$$

$$E \left[\begin{bmatrix} \mathbf{w}(t) \\ \dots \\ \mathbf{w}_f(t) \end{bmatrix} \begin{bmatrix} \mathbf{w}^T(t+\tau) & \vdots & \mathbf{w}_f^T(t+\tau) \end{bmatrix} \right] = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & q\mathbf{Q}_f \end{bmatrix}$$

where q is a scalar and \mathbf{Q}_f is an arbitrary positive definite matrix. This procedure, in effect, makes the control system look like an LQR asymptotically as $q \rightarrow \infty$ by reducing the filter's reliance on the control input and making the filter faster so that its dynamics can be effectively ignored. The LQR-like robustness properties can only be attained using this procedure if the number of measurements is greater than or equal to the number of control inputs and the plant transfer function $\mathbf{H}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ is minimum phase. An alternative LTR procedure consists of modifying the cost function's state weighting matrix to increase robustness (8). This procedure can be implemented when the number of measurements is less than or equal to the number of control inputs. A measurement weighting term is

added to the cost function as (8)

$$\begin{aligned} \bar{J}_{LQG}(\hat{\mathbf{x}}(t_i^+), \mathbf{u}(t_i)) &= \frac{1}{2}E \left[\lim_{N \rightarrow \infty} \frac{1}{t_N} \sum_{i=0}^N (\hat{\mathbf{x}}^T(t_i^+) (\mathbf{X} + q_{LTR} \mathbf{H}^T \mathbf{H}) \hat{\mathbf{x}}(t_i^+) \right. \\ &\quad \left. + \mathbf{u}^T(t_i) \mathbf{U} \mathbf{u}(t_i)) \right] \end{aligned} \quad (3.44)$$

The value of q_{LTR} is adjusted to attain the desired robustness. This procedure is also restricted such that the plant transfer function must be minimum phase. However, the two procedures just described may also work for non-minimum phase systems, but there is no guarantee of the effectiveness of LTR in this case.

3.2.3.1 Stability. The closed loop system consisting of the LQG controller and the plant (described by the same plant model used in the LQG design) is asymptotically stable when both the Kalman filter and LQR designs are stable. Therefore the LQG design is stable if the Kalman filter stability criteria and LQR stability criteria are met (reference (11) proves that these are sufficient conditions for LQG stability).

With a time-invariant plant model (this research pertains to a time-invariant system), an LQR system is stable if the plant model is stabilizable from the points of entry of $\mathbf{u}(t)$ and detectable from the points of extraction of $\mathbf{y}_a(t)$, where $\mathbf{y}_a(t)$ is called the artificial output and $\mathbf{y}_a(t) = \sqrt{\mathbf{W}_{xx}}^T \mathbf{x}(t)$, where \mathbf{W}_{xx} is the state weighting matrix from the cost function (11). A system model is stabilizable if its unstable subspace is contained in its controllable subspace; or in other words, all unstable modes of the system (if there are any) are controllable. The controllable subspace is defined by the controllability matrix

$$\mathbf{M}_C = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \dots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}$$

where n is the number of states in the plant model. The rank of \mathbf{M}_C is the number of controllable states, therefore, in order for the system to be stabilizable, any state that represents an unstable mode must be a controllable state. Similarly, a system model is detectable if its unobservable subspace is contained in its stable subspace; or in other words, all unstable modes of the system (if there are any) are observable. The observable

subspace is defined by the observability matrix

$$\mathbf{M}_O = \begin{bmatrix} \sqrt{\mathbf{W}_{xx}}^T & : & \mathbf{A}^T \sqrt{\mathbf{W}_{xx}}^T & : & \dots & : & (\mathbf{A}^T)^{n-1} \sqrt{\mathbf{W}_{xx}}^T \end{bmatrix}$$

The rank of \mathbf{M}_O is the number of observable states, therefore, in order for the system to be detectable, any state that represents an unstable mode must be an observable state. The controllability and observability matrices associated with the equivalent discrete-time model are identical to those just presented with \mathbf{A} replaced by Φ , \mathbf{B} by \mathbf{B}_d , and \mathbf{W}_{xx} by \mathbf{X} :

$$\mathbf{M}_C = \begin{bmatrix} \mathbf{B}_d & : & \Phi \mathbf{B}_d & : & \dots & : & \Phi^{n-1} \mathbf{B}_d \end{bmatrix} \quad (3.45)$$

$$\mathbf{M}_O = \begin{bmatrix} \sqrt{\mathbf{X}}^T & : & \Phi^T \sqrt{\mathbf{X}}^T & : & \dots & : & (\Phi^T)^{n-1} \sqrt{\mathbf{X}}^T \end{bmatrix} \quad (3.46)$$

A Kalman filter is stable if the plant model is stabilizable from the points of entry of $\mathbf{w}(t)$ and detectable from the points of extraction of $\mathbf{z}(t)$. In this case, the controllability matrix associated with the Kalman filter is

$$\mathbf{M}_C = \begin{bmatrix} \mathbf{G} & : & \mathbf{A} \mathbf{G} & : & \dots & : & \mathbf{A}^{n-1} \mathbf{G} \end{bmatrix}$$

and the observability matrix is

$$\mathbf{M}_O = \begin{bmatrix} \mathbf{H}^T & : & \mathbf{A}^T \mathbf{H}^T & : & \dots & : & (\mathbf{A}^T)^{n-1} \mathbf{H}^T \end{bmatrix}$$

Finally, for the equivalent discrete-time model, replacing \mathbf{A} by Φ , and \mathbf{G} with \mathbf{G}_d , the controllability and observability matrices are

$$\mathbf{M}_C = \begin{bmatrix} \mathbf{G}_d & : & \Phi \mathbf{G}_d & : & \dots & : & \Phi^{n-1} \mathbf{G}_d \end{bmatrix} \quad (3.47)$$

$$\mathbf{M}_O = \begin{bmatrix} \mathbf{H}^T & : & \Phi^T \mathbf{H}^T & : & \dots & : & (\Phi^T)^{n-1} \mathbf{H}^T \end{bmatrix} \quad (3.48)$$

3.2.4 Summary. Discussion up to this point has been limited to describing the LQG controller as a regulator that drives the plant states to zero. However, small modifications can be made to the design to yield an LQG tracker to track a desired reference

input(s) (11). The following sections of this chapter layout the specific LQG design implemented in this research.

3.3 Controller Design

Figure 3.1 outlines the main components of the general control loop. The overall plant consists of four sub-plants: CSM azimuth axis, FSM azimuth axis, CSM elevation axis, FSM elevation axis. As mentioned in Chapter 2, the average deflection of the wavefront of light away from nominal at the aperture of the telescope can be broken into two components: tilt (azimuth axis) and tip (elevation axis). Tilt and azimuth, and tip and elevation, will be used interchangeably to describe characteristics and properties of the light, mirrors, sensor, and controller. Tilt and tip in the context of this research describes the angular deflection from nominal in nanoradians. Therefore, the input commands from the control to the sub-plants (one input per sub-plant) result in an angular deflection of the mirror about the appropriate axis. An angular deflection of k nrads by a mirror about one axis results in inducing $2k$ nrads of tilt on the same axis of a wavefront propagating off the mirror (6). However, as mentioned in Section 1.2, in this research the factor of two is ignored so that k nrads from the mirror induces the same k nrads of tilt on the wavefront. If the design is eventually implemented in the actual AO system, a simple gain will be needed to account for the simplification. Figure 3.2 presents the optical path of the light as it arrives and propagates off the steering mirrors in series. The individual outputs of the sub-plants represent the angular deflection of the sub-plants, which corresponds to the additional tilt and tip induced on the propagating wavefront by the mirrors (6). In Figure 3.1, the output of the CSM azimuth axis is summed with the output of the FSM azimuth axis, and the same applies to the elevation axis. That summation represents the fact that the wavefront propagates off the CSM and FSM in series and is equal to the total induced tilt on the wavefront. The output of the mirrors is then summed with the tilt disturbance. The tilt disturbance represents the actual tilt and tip present on the wavefront after the light enters the telescope and just prior to the light propagating off the mirrors. Therefore, the latter summation represents the tilt and tip present on the wavefront prior to reaching the steering mirrors plus the output of the steering mirrors. The pyramid sensor provides

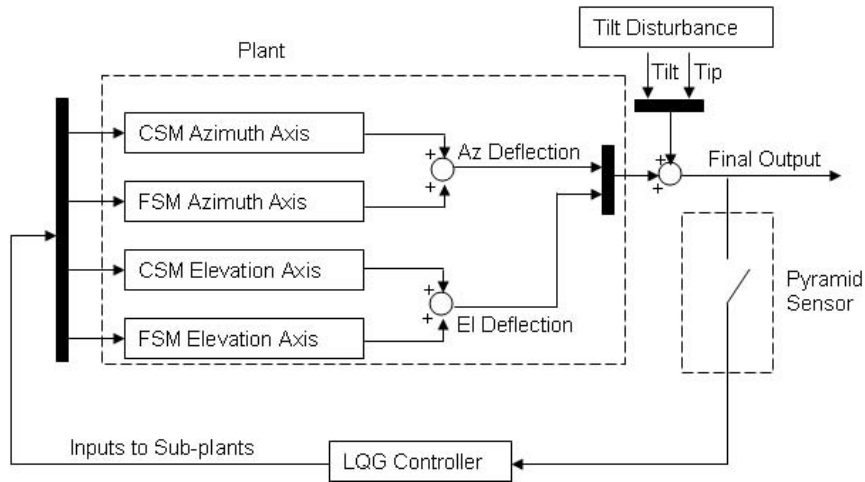


Figure 3.1 Control Loop

noisy measurements of the wavefront tilt and tip after the light has propagated off the mirrors (represented as the final output in Figure 3.1) at a 5 kHz rate, which equates to a sample period of 0.2 msec (6). The primary goal of the control system is to drive the final output to zero so that the wavefront of light has no tilt and no tip after propagating off both mirrors. This implies that the controller's function is to ensure that the output of the plant equals the negative of the tilt disturbance (rather than zero as is the function of a regulator), resulting in the need for an LQG tracker. The LQG regulator can be easily extended to take on the role of a tracker in that now the tracking error (in this case tilt and tip) needs to be regulated to zero.

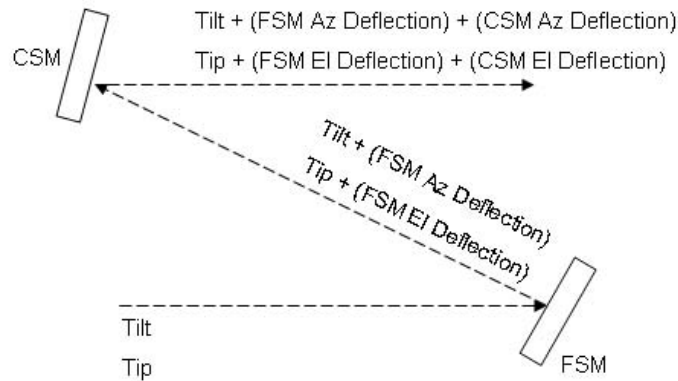


Figure 3.2 Optical Path Through the Steering Mirrors

3.3.1 Design Parameters. Table 3.1 lists the parameters of the system and environment used in this research. The Greenwood frequencies are provided by the SOR as representative values for Kirtland Air Force Base (6).

Table 3.1 Design Parameters

Parameter	Value
Aperture Diameter, d	3.5 meters
Wavelength of Light, λ	0.5 μm
Fried Radius, r_0	5 cm
Sample Time, T_s	0.2 ms
Minimum Greenwood Frequency, $f_{G_{\min}}$	42 Hz
Average Greenwood Frequency, $f_{G_{avg}}$	88 Hz
Maximum Greenwood Frequency, $f_{G_{\max}}$	191 Hz

3.3.2 Kalman Filter Model. The pyramid sensor provides noisy measurements of the output of the system, and the output of the system will be seen to be a linear combination of the plant and disturbance states; therefore, a Kalman filter is designed to be cascaded with the LQR tracker to form the LQG controller. Applying the separation principle, the Kalman filter is designed independent of the controller, with the sole objective of estimating the plant and disturbance states given inputs from the controller and the pyramid sensor measurements. The following sections develop the Kalman filter model.

3.3.2.1 FSM and CSM Models. Initial linear dynamics models of the CSM and FSM were provided by Dr. James Brown of the SOR for use in this research (6). These transfer function models of the CSM and FSM were estimated based on frequency response data attained experimentally, which were also provided by the SOR for this research. As previously mentioned in Section 1.2, one CSM dynamics model was provided from the SOR to represent the dynamics of both the azimuth and elevation axes. The actual CSM frequency response data from each axis are shown in Figure 3.3. The data indicate that the dynamics of the two axes are nearly identical, further justifying the use of the same model for both axes. The frequency response data and the estimated transfer function model are shown in Figures 3.4 and 3.5. The frequency response data are noisy near 10^3

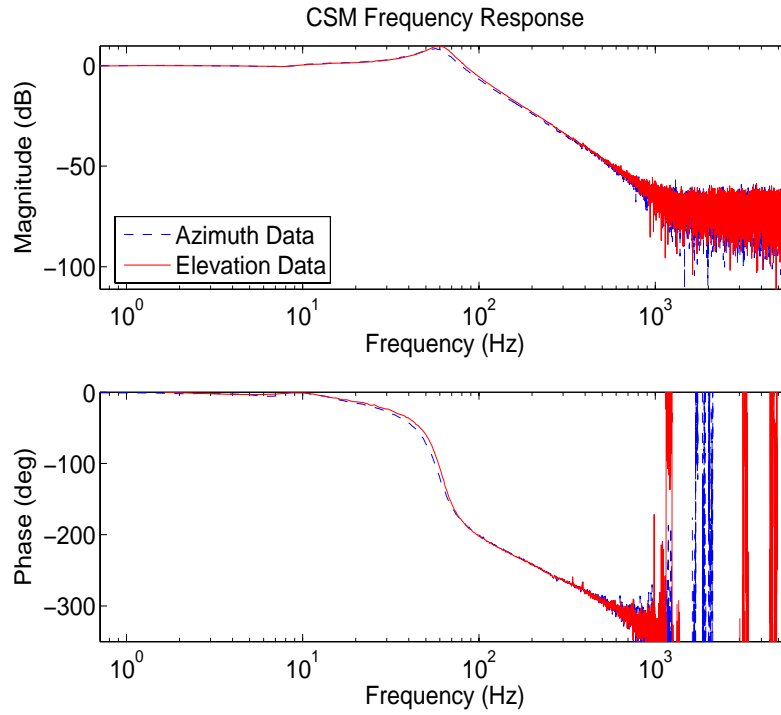


Figure 3.3 CSM Frequency Response Data

Hz and above. The model fits the data very well except near the noisy data, which cannot be trusted. The provided transfer function is

$$G_{CSM}^{Az/El} = \frac{1.497E8}{s^3 + 1137s^2 + (2.799E5)s + 1.497E8} \quad (3.49)$$

with poles and zeros shown in Table 3.2.

Table 3.2 CSM Transfer Function Poles and Zeros

CSM Poles	CSM Zeros
-1006.7	None
$-65.2 \pm 380.1j$	

Two separate sets of FSM frequency response data were provided: one set developed from the response of the azimuth axis and the other from the elevation axis. These two data sets are shown in Figure 3.6 and do differ from each other in certain frequency

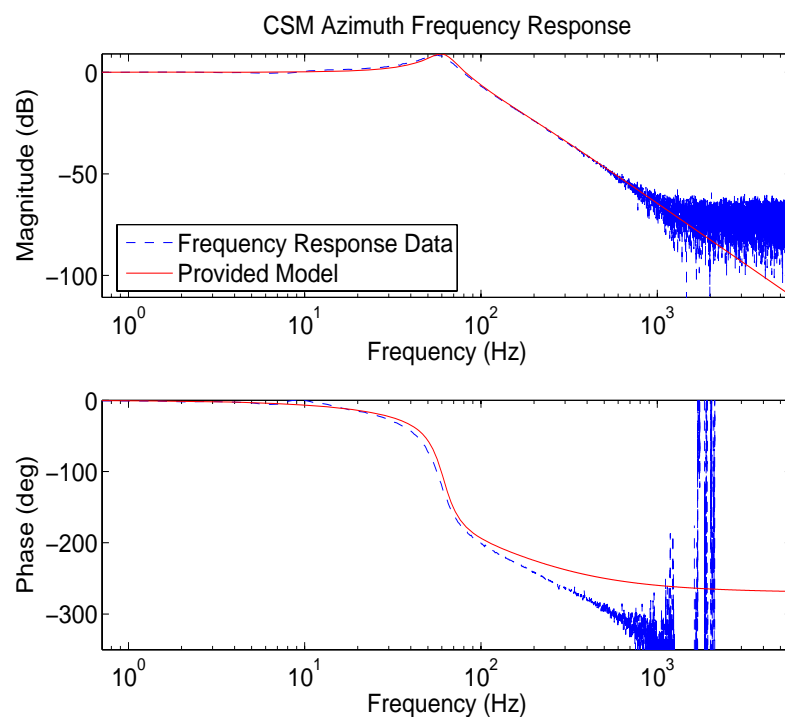


Figure 3.4 Provided CSM Azimuth Model versus the Frequency Response Data

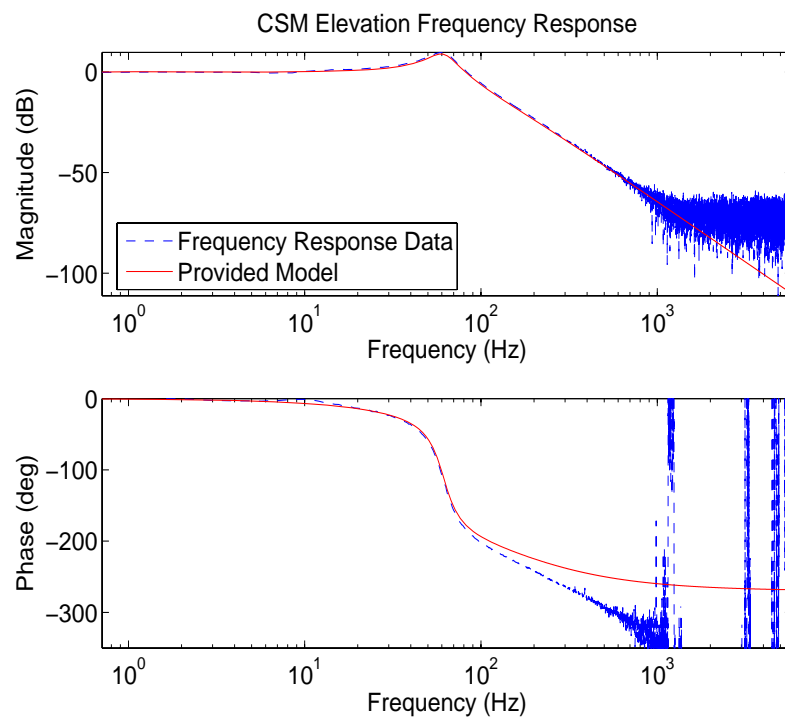


Figure 3.5 Provided CSM Elevation Model versus the Frequency Response Data

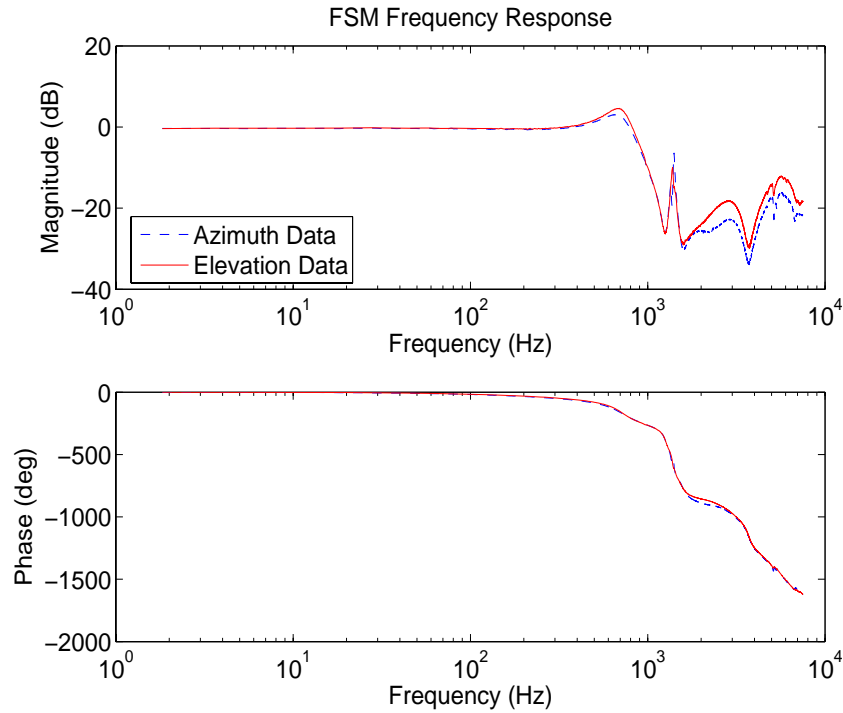


Figure 3.6 FSM Frequency Response Data

bands. However, one transfer function model was provided from the SOR to represent the dynamics of both the azimuth and elevation axes. It is evident from the magnitude plots of Figure 3.7 and 3.8 that the transfer function was developed to match the actual dynamics of the elevation axis. To develop a more accurate model of the FSM elevation axis, a transfer function-fitting MATLAB[®] algorithm, called *invfreqs*, was used (10). The algorithm was also used to develop a more accurate model of the azimuth axis. The algorithm provides a least-squares fit to magnitude and phase frequency response data resulting in a transfer function of specified order (i.e., number of poles and zeros). This function allows the allotment of different weighting to specific frequencies for the curve-fit. This frequency weighting was particularly useful given the highly nonlinear characteristics of the FSM at frequencies of 10^3 and 10^4 Hz. Several transfer functions were developed for both axes using this method, varying in the number of poles and zeros. Tables 3.3 and 3.4 show comparisons of the error statistics of each transfer function relative to the frequency response data. The mean error was calculated as the *average absolute difference* between the magnitude of the modeled transfer function and the experimental magnitude across

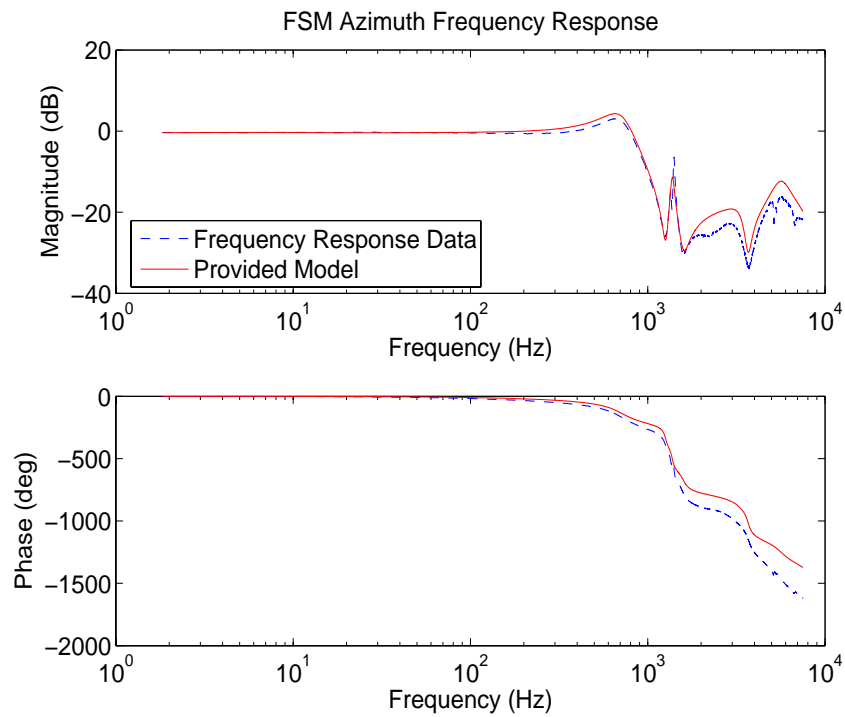


Figure 3.7 Provided FSM Azimuth Model versus the Frequency Response Data

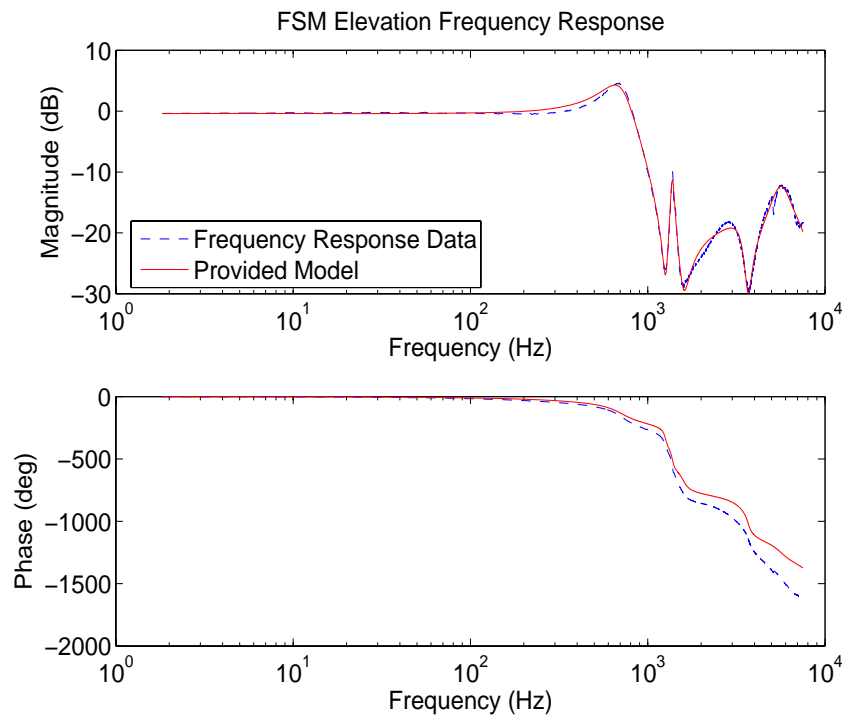


Figure 3.8 Provided FSM Elevation Model versus the Frequency Response Data

all frequency data values provided in the data sets. The variance is centered about the mean error as just defined. The magnitude of the transfer functions is unit-less; therefore the mean error and variance are as well. The transfer functions labeled as number 3

Table 3.3 FSM Azimuth Transfer Function Error Statistics

FSM Azimuth Axis		
Transfer Function	Mean Error	Variance
1. Provided Model (12 th Order)	3.2146	2.7597
2. 9 th Order	2.0380	3.6505
3. 10th Order	0.9825	1.1436
4. 11 th Order	0.7881	0.8859
5. 12 th Order	0.9545	1.0642
6. 13 th Order	0.8546	1.0649
7. 14 th Order	0.7837	0.7353
8. 15 th Order	0.5677	0.3310
9. 18 th Order	0.2773	0.1571

Table 3.4 FSM Elevation Error Statistics

FSM Elevation Axis		
Transfer Function	Mean Error	Variance
1. Provided Model (12 th Order)	0.5968	0.1685
2. 10 th Order	2.1657	3.6079
3. 11th Order	0.5058	0.3119
4. 18 th Order	0.2104	0.0612

(from Tables 3.3 and 3.4) were chosen based on having a low number of poles (and thus less complexity in the eventual plant model used in the controller design) and yet still maintaining a good level of accuracy, as compared to the higher order transfer functions. Emphasis was placed on low order models for the selection of the transfer functions, and the chosen transfer functions, which are able to represent all of the dynamic modes, show the most drastic improvement over the next lowest models. Transfer function 4, in Table 3.3, could have been chosen based on drastically improved statistics over transfer function 5 and better statistics than transfer function 3; however, transfer function 3 was chosen

due to its lower order. Figures 3.9 to 3.14 show how several of the transfer functions from Tables 3.3 and 3.4 compare to the data. Transfer functions of lower order than the two chosen have significantly degraded error statistics as a result of the inability to represent all of the major dynamic modes. Both of the chosen transfer functions are lower order than the original model provided by SOR and have better or similar error statistics.

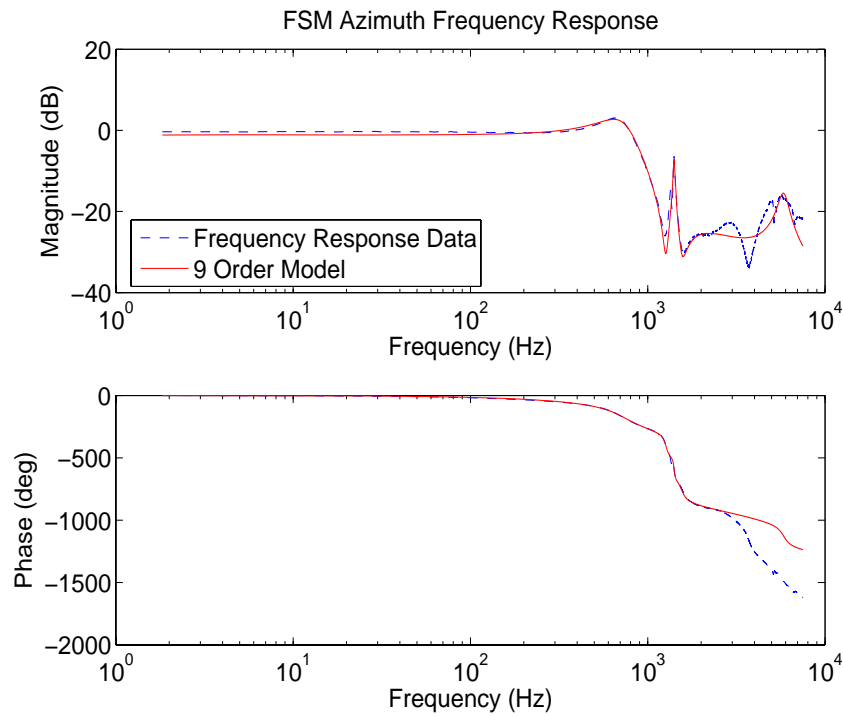


Figure 3.9 Azimuth Model #2

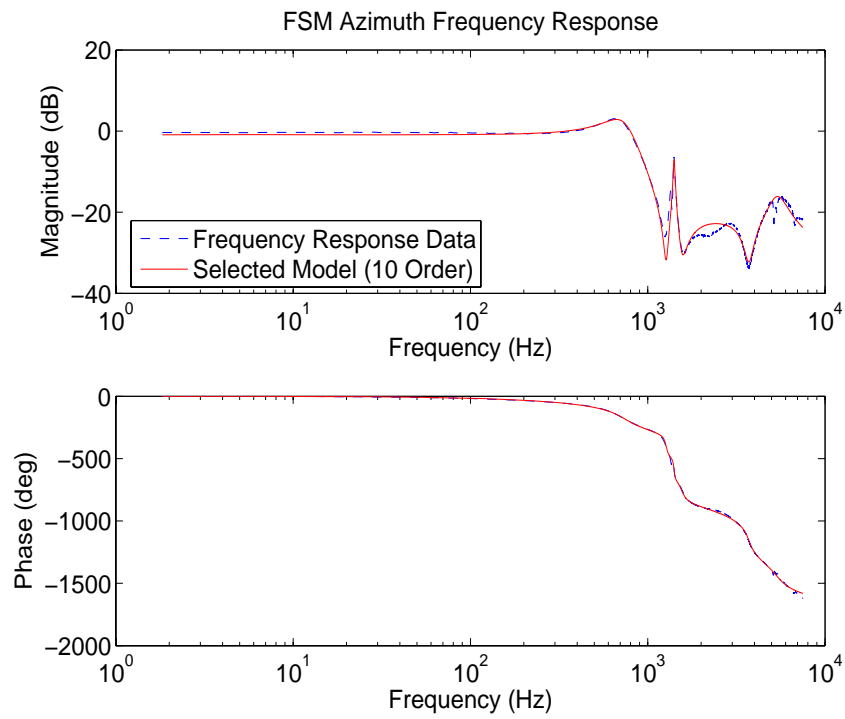


Figure 3.10 Azimuth Model #3

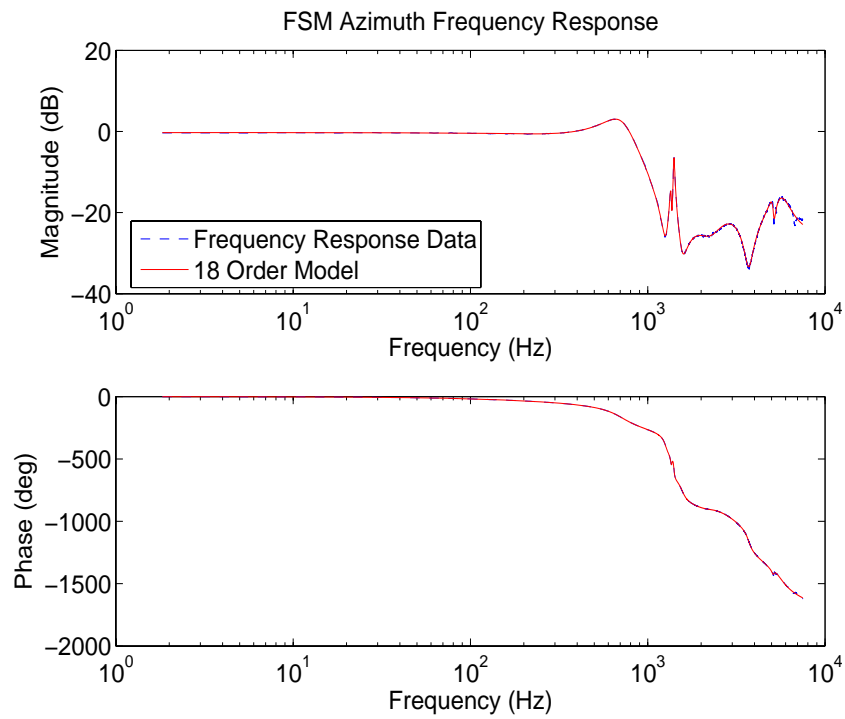


Figure 3.11 Azimuth Model #9

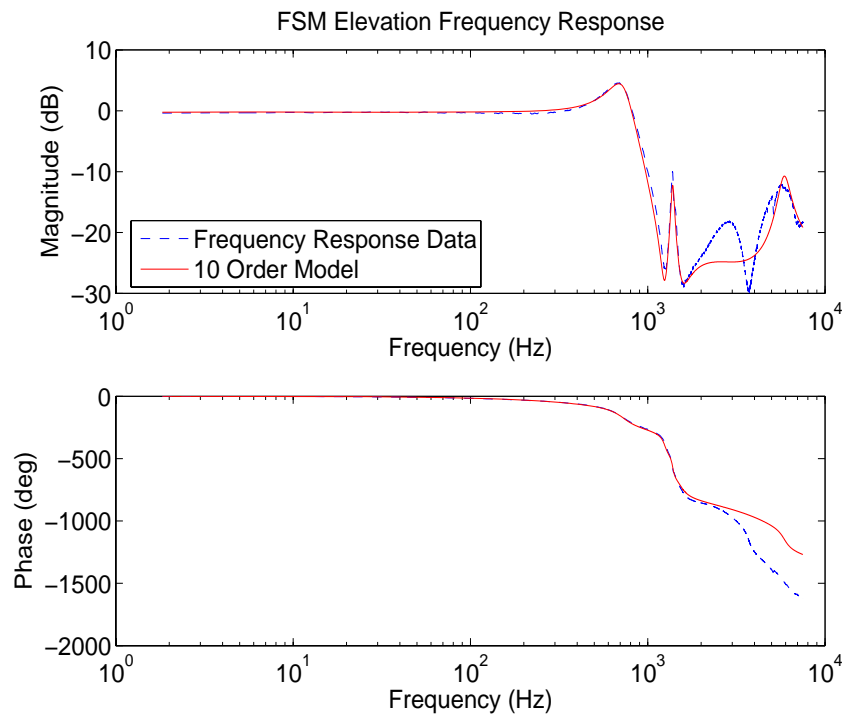


Figure 3.12 Elevation Model #2

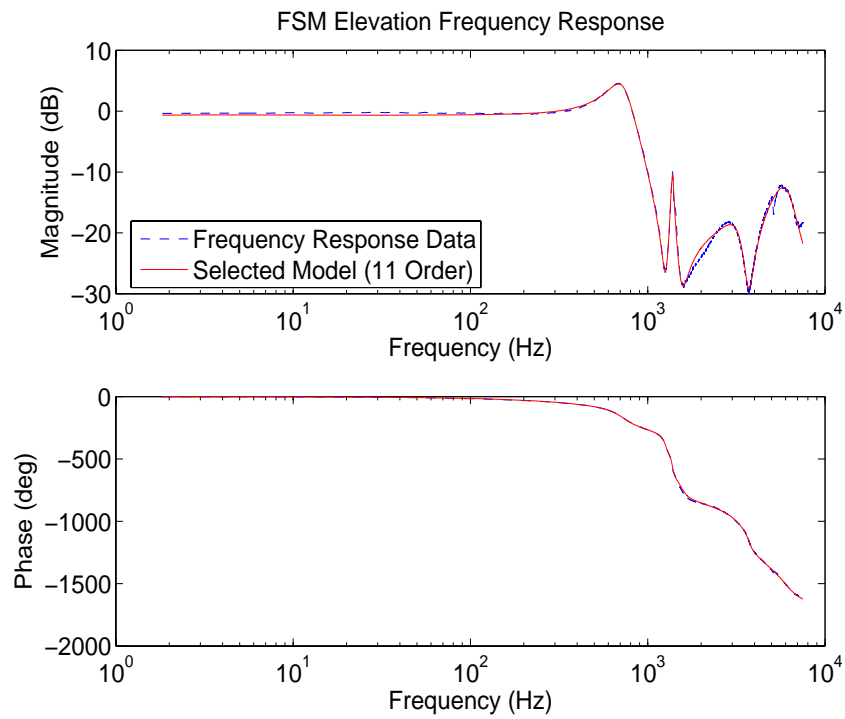


Figure 3.13 Elevation Model #3

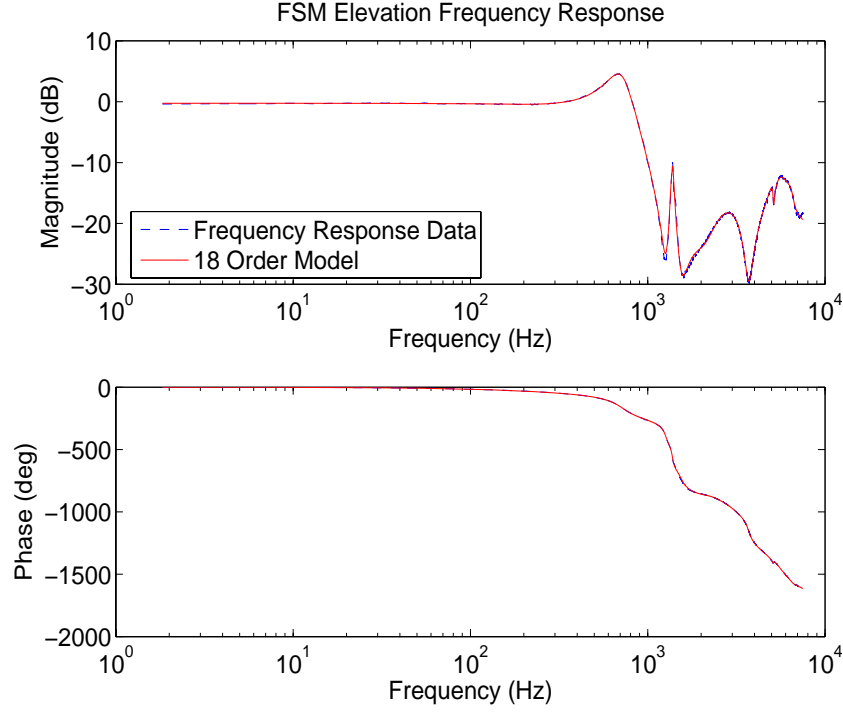


Figure 3.14 Elevation Model #4

The resulting transfer functions used to model the FSM are

$$G_{FSM}^{Az} = \frac{\left(\begin{aligned} &-2133s^9 + (9.409E7)s^8 - (4.606E12)s^7 + (1.033E17)s^6 - (2.381E21)s^5 \\ &+ (3.229E25)s^4 - (3.088E29)s^3 + (3.216E33)s^2 \\ &- (1.128E37)s + 9.767E40 \end{aligned} \right)}{\left(\begin{aligned} &s^{10} + (4.144E4)s^9 + (2.433E9)s^8 + (6.381E13)s^7 + (1.475E18)s^6 \\ &+ (2.336E22)s^5 + (2.217E26)s^4 + (1.915E30)s^3 \\ &+ (1.071E34)s^2 + (3.555E37)s + 1.085E41 \end{aligned} \right)} \quad (3.50)$$

$$G_{FSM}^{El} = \frac{\left(\begin{aligned} &-(5.193E7)s^9 + (3.437E12)s^8 - (1.569E17)s^7 + (4.259E21)s^6 \\ &-(9.141E25)s^5 + (1.494E30)s^4 - (1.259E34)s^3 + (1.547E38)s^2 \\ &- (4.776E41)s + 4.731E45 \end{aligned} \right)}{\left(\begin{aligned} &s^{11} + (4.099E4)s^{10} + (3.581E9)s^9 + (9.124E13)s^8 + (3.615E18)s^7 \\ &+ (5.348E22)s^6 + (1.137E27)s^5 + (8.937E30)s^4 + (8.785E34)s^3 \\ &+ (4.660E38)s^2 + (1.574E42)s + 5.099E45 \end{aligned} \right)} \quad (3.51)$$

consisting of the poles and zeros listed in Tables 3.5 and 3.6 Both FSM transfer functions

Table 3.5 FSM Azimuth Axis Poles and Zeros

Poles	Zeros
$-4441 \pm 33514j$	$12736 \pm 27483j$
$-4328 \pm 22893j$	$1231 \pm 23364j$
-14720	14857
$-97 \pm 8850j$	$470 \pm 9797j$
-7021	$196 \pm 7986j$
$-985 \pm 4542j$	

Table 3.6 FSM Elevation Axis Poles and Zeros

Poles	Zeros
$-5408 \pm 38009j$	$18659 \pm 27994j$
$-8060 \pm 29995j$	$1167 \pm 23329j$
-5243	24648
$-3472 \pm 20541j$	$637 \pm 9727j$
$-775 \pm 4518j$	$307 \pm 7931j$
$-157 \pm 8659j$	

are nonminimum-phase since all of the zeros are in the right half of the real and imaginary plane. The nonminimum-phase characteristic of the transfer functions may be the result of unknown time delays present in the FSM frequency response data.

The azimuth and elevation axes of both the CSM and FSM are considered together as the overall plant to be controlled. A state-space model of the entire plant can be formed from the transfer functions of the mirrors. In general, the state-space model will be of the

form

$$\begin{aligned} \dot{\mathbf{x}}_{Plant}(t) = & \begin{bmatrix} \mathbf{A}_{CSM}^{Az} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{CSM}^{El} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{FSM}^{Az} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{FSM}^{El} \end{bmatrix} \mathbf{x}_{Plant}(t) \\ & + \begin{bmatrix} \mathbf{B}_{CSM}^{Az} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{CSM}^{El} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{FSM}^{Az} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{FSM}^{El} \end{bmatrix} \mathbf{u}(t) \end{aligned} \quad (3.52)$$

$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{C}_{CSM}^{Az} & \mathbf{0} & \mathbf{C}_{FSM}^{Az} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{CSM}^{El} & \mathbf{0} & \mathbf{C}_{FSM}^{El} \end{bmatrix} \mathbf{x}_{Plant}(t) \quad (3.53)$$

Notice that coupling between the sub-plants is present in Equation (3.53). The transfer functions are converted to the standard controllable state-space model. The matrices of the standard controllable model contain individual elements equal to the polynomial coefficients of the CSM and FSM transfer functions in Equations (3.49), (3.50), and (3.51). The coefficients of the FSM transfer functions have extremely large orders of magnitude in variation (up to 40 orders of magnitude) that are present in the standard controllable model matrices. To avoid potential numerical problems in simulation and implementation, the state-space system was put into a modified Jordan canonical form by the transformation equations (13)

$$\mathbf{A}_{canonical} = \mathbf{T}^{-1} \mathbf{A} \mathbf{T}$$

$$\mathbf{B}_{canonical} = \mathbf{T}^{-1} \mathbf{B}$$

$$\mathbf{C}_{canonical} = \mathbf{C} \mathbf{T}$$

$$\mathbf{D}_{canonical} = \mathbf{D}$$

The columns of \mathbf{T} are the real and imaginary parts of the complex eigenvectors of \mathbf{A} . The modified Jordan canonical form results in real-valued system matrices, despite the complex

eigenvalues of \mathbf{A} , with much lower elemental magnitude differences than the standard controllable model matrices (see Appendix A.1.1).

3.3.2.2 Tilt Disturbance Model. The actual tilt and tip on the wavefront induced by the turbulent atmosphere is modeled using a shaping filter driven by WGN. This is well represented by an exponentially time-correlated process model, so the shaping filter is a first order lag driven by zero-mean WGN of strength $\mathbf{Q}_{disturb}$. The output of the model has autocorrelation

$$E[x_{disturb}(t)x_{disturb}(t + \tau)] = \sigma^2 e^{-|\tau|/T}$$

where $x_{disturb}$ is the tilt (or tip), σ^2 is the variance of the tilt (or tip), and T is the correlation time of tilt (or tip). This model is an approximation of the tilt/tip stochastic process, but is applicable since the tilt/tip disturbance process is zero-mean, of a certain variance, and time-correlated in an approximately exponential fashion. The state equation of the model is

$$\dot{\mathbf{x}}_{disturb}(t) = \mathbf{A}_{disturb}\mathbf{x}_{disturb}(t) + \mathbf{w}_{disturb}(t)$$

where $\mathbf{x}_{disturb}(t)$ is a vector of two states, tilt and tip, $\mathbf{w}_{disturb}(t)$ is zero-mean WGN of strength

$$\mathbf{Q}_{disturb} = \begin{bmatrix} 2\frac{\sigma_{tilt}^2}{T_{tilt}} & 0 \\ 0 & 2\frac{\sigma_{tip}^2}{T_{tip}} \end{bmatrix}$$

and

$$\mathbf{A}_{disturb} = \begin{bmatrix} \frac{-1}{T_{tilt}} & 0 \\ 0 & \frac{-1}{T_{tip}} \end{bmatrix} \quad (3.54)$$

This model is based on prior research performed by David Anderson and Mark Van Bokern in which the model was used for the same purpose (3, 18). Anderson presented correlation time values derived by Von Bokern from first-order Markov process functions that were fit to autocorrelation data. An aperture diameter of 1.5 meters was used to develop the autocorrelation data (rather than 3.5 meters as used in this research), however, the correlation time values are still used as a close approximation. The correlation times are

based on an average atmospheric wind speed of 21 m/s and are (3)

$$T_{tilt} = 0.32 \text{ sec}$$

$$T_{tip} = 0.32 \text{ sec}$$

thus

$$\mathbf{A}_{disturb} = \begin{bmatrix} -3.125 & 0 \\ 0 & -3.125 \end{bmatrix}$$

from Equation (3.54). The variance values used in the model are derived from the variance of the tilt and tip Zernike coefficients. The variance of the Zernike coefficients is in units of radians of phase (squared), which is converted to angular nrad (squared) by the following procedure. The angular tilt and tip $\theta_{tilt/tip}$ is determined by

$$\theta_{tilt/tip} = \frac{(\frac{2a_{tilt/tip}}{2\pi})\lambda(10^9)}{d/2} \quad (3.55)$$

where $a_{tilt/tip}$ is the tilt or tip Zernike coefficient, a_2 or a_3 , and the scale factor of 10^9 converts the units from angular radians to angular nanoradians. Figure 3.15 aids in the understanding of the following explanation of Equation (3.55). From Equations (3.7) and (3.10), the maximum phase of tilt within the aperture is found when $\theta = 0$ and $r = 1$ (the edge of the aperture), thus

$$\phi_{tilt}^{\max} = 2a_{tilt}$$

Similarly, the maximum phase of tip comes from Equations (3.8) and (3.11) when $\theta = \frac{\pi}{2}$ and $r = 1$, resulting in

$$\phi_{tip}^{\max} = 2a_{tip}$$

Considering that one wavelength is 2π radians of phase, $\frac{\phi_{tilt/tip}^{\max}}{2\pi}$ is the number of wavelengths contained in the maximum phase. Therefore, $\frac{\phi_{tilt/tip}^{\max}}{2\pi}\lambda$ is the physical distance in meters from the nominal wavefront to the tilted wavefront at the edge of the aperture.

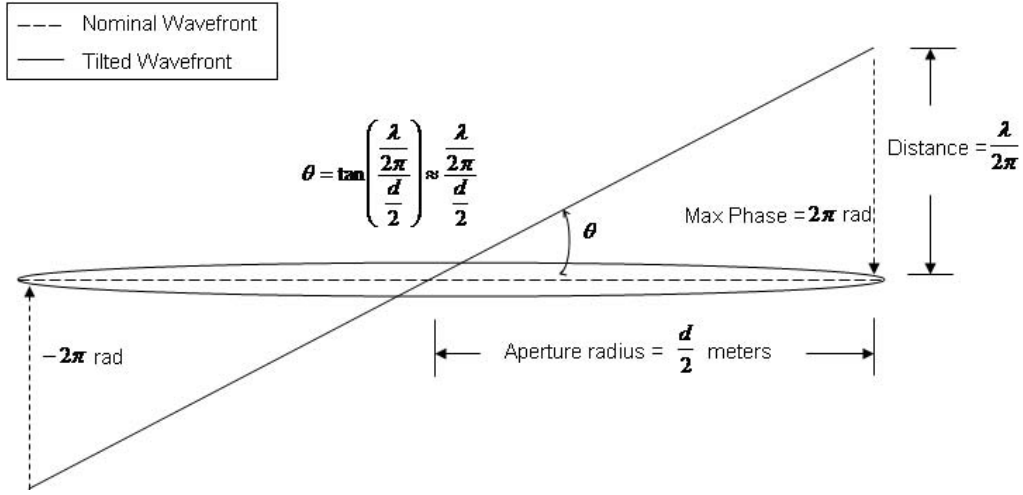


Figure 3.15 Example of determining the angular tilt using 2π maximum phase

Knowing the radius of the aperture, the angular tilt and tip can be found by

$$\theta_{tilt/tip} = \tan \frac{(\frac{2a_{tilt/tip}}{2\pi})\lambda}{d/2} \quad (3.56)$$

Equation (3.55) results from applying a small angle approximation, $\tan \theta_{small} \approx \theta_{small}$, and the 10^9 conversion factor to Equation (3.56). The variance of the angular tilt and tip, which is needed to define $\mathbf{Q}_{disturb}$, is

$$E[\theta_{tilt/tip}^2] = E \left[\left(\frac{(\frac{2a_{tilt/tip}}{2\pi})\lambda(10^9)}{d/2} \right)^2 \right] = \frac{\lambda^2(10^9)^2}{\pi^2(d/2)^2} E[a_{tilt/tip}^2] \quad (3.57)$$

where $E[a_{tilt/tip}^2]$ is the variance of the Zernike coefficient defined by Equation (3.9). Applying the parameters in Table 3.1 to Equations (3.9) and (3.57) results in

$$\sigma_{tilt}^2 = 4.4056E6 \text{ nrad}^2$$

$$\sigma_{tip}^2 = 4.4056E6 \text{ nrad}^2$$

$$\mathbf{Q}_{disturb} = \begin{bmatrix} 2.7535E7 \frac{\text{nrad}^2}{\text{sec}} & 0 \\ 0 & 2.7535E7 \frac{\text{nrad}^2}{\text{sec}} \end{bmatrix}$$

3.3.2.3 *Final Kalman Filter Form.* The final Kalman filter model shown in Equations (3.58) and (3.59) results from augmenting the plant model with the tilt disturbance model as well as adding WGN defined by Equation (3.20) to the dynamics to represent potential mismodeling:

$$\begin{aligned}
 \begin{bmatrix} \dot{\mathbf{x}}_{Plant}(t) \\ \dot{\mathbf{x}}_{disturb}(t) \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_{CSM}^{Az} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{CSM}^{El} & \mathbf{0} & \mathbf{0} & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{FSM}^{Az} & \mathbf{0} & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{FSM}^{El} & \vdots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \vdots & \mathbf{A}_{disturb} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{Plant}(t) \\ \mathbf{x}_{disturb}(t) \end{bmatrix} \\
 &+ \begin{bmatrix} \mathbf{B}_{CSM}^{Az} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{CSM}^{El} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{FSM}^{Az} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{FSM}^{El} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{u}(t) + \begin{bmatrix} \mathbf{w}(t) \\ \mathbf{w}_{disturb}(t) \end{bmatrix} \quad (3.58)
 \end{aligned}$$

$$\mathbf{z}(t_i) = \begin{bmatrix} \mathbf{C}_{CSM}^{Az} & \mathbf{0} & \mathbf{C}_{FSM}^{Az} & \mathbf{0} & \vdots & 1 & 0 \\ \mathbf{0} & \mathbf{C}_{CSM}^{El} & \mathbf{0} & \mathbf{C}_{FSM}^{El} & \vdots & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{Plant}(t_i) \\ \mathbf{x}_{disturb}(t_i) \end{bmatrix} + \mathbf{v}(t_i) \quad (3.59)$$

where

$$\begin{aligned}
 \mathbf{Q} &= \begin{bmatrix} \mathbf{Q}_{CSM}^{Az} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{CSM}^{El} & \mathbf{0} & \mathbf{0} & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{FSM}^{Az} & \mathbf{0} & \vdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{FSM}^{El} & \vdots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \vdots & \mathbf{Q}_{disturb} \end{bmatrix} \\
 \mathbf{R} &= \begin{bmatrix} R_{Az} & 0 \\ 0 & R_{El} \end{bmatrix}
 \end{aligned}$$

The measurement model represents the summation of the plant output and the tilt disturbance in each axis plus a WGN noise defined by Equation (3.32) to represent the pyramid sensor noise. The \mathbf{R} , \mathbf{Q}_{CSM}^{Az} , \mathbf{Q}_{CSM}^{El} , \mathbf{Q}_{FSM}^{Az} , and \mathbf{Q}_{FSM}^{El} matrix elements have been left as variables to be tuned iteratively during simulations to determine the values that provide best performance. The augmented dynamics model is then discretized using Equations (3.22) to (3.25) with a sample period of 0.2 msec, resulting in the state-space difference equation matrices shown in Appendix A.2.1. The initial state estimate vector used is

$$\begin{bmatrix} \hat{\mathbf{x}}_{0_{Plant}}(t) \\ \hat{\mathbf{x}}_{0_{disturb}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

The steady-state Kalman filter gain $\bar{\mathbf{K}}$ found by Equations (3.40) and (3.41), with Φ , \mathbf{Q}_d , \mathbf{H} , and \mathbf{R} as just defined, is used in this design. The state estimates are determined at each sample time by Equations (3.33) and (3.35) with Φ , \mathbf{B}_d , \mathbf{H} , and $\bar{\mathbf{K}}$ as just defined.

3.3.3 LQG/LTR Tracker Design. The desire is to design an LQG tracker that provides zero-mean steady-state error while tracking a desired signal \mathbf{y}_d . This is accomplished through the use of a Proportional-plus-Integral (PI) LQG tracker based on the pseudointegral of the regulation error (11). The regulation error is

$$\mathbf{y}_c(t_i) - \mathbf{y}_d = \mathbf{H}\mathbf{x}(t_i) - \mathbf{y}_d$$

where \mathbf{y}_c is the controlled output, $\mathbf{H}\mathbf{x}$, which represents the wavefront tilt and tip after correction has been applied by the CSM and FSM. The desired signal is a vector of zeros, $\mathbf{y}_d = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$, since the desired outcome is zero tilt and tip on the wavefront, and \mathbf{H} is from the Kalman filter model, Equation (3.59). The controller design consists of two additional pseudointegral error states defined as

$$\mathbf{e}_I(t_{i+1}) = \mathbf{e}_I(t_i) + [\mathbf{y}_c(t_i) - \mathbf{y}_d] = \mathbf{e}_I(t_i) + \mathbf{y}_c(t_i)$$

since $\mathbf{y}_d = \mathbf{0}$. The pseudointegral states are augmented to the discrete-time dynamics model used by the Kalman filter to serve as the tracker's dynamics model (the Kalman

filter does not estimate the pseudointegral states) shown in Equation (3.60):

$$\begin{aligned}
\begin{bmatrix} \mathbf{x}_{Plant}(t_{i+1}) \\ \mathbf{x}_{disturb}(t_{i+1}) \\ \mathbf{x}_{e_I}(t_{i+1}) \end{bmatrix} &= \begin{bmatrix} \Phi_{CSM}^{Az} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \mathbf{0} & \Phi_{CSM}^{El} & \mathbf{0} & \mathbf{0} & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \Phi_{FSM}^{Az} & \mathbf{0} & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Phi_{FSM}^{El} & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & \Phi_{disturb}^{1,1} & 0 & \vdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \vdots & 0 & \Phi_{disturb}^{2,2} & \vdots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{C}_{CSM}^{Az} & \mathbf{0} & \mathbf{C}_{FSM}^{Az} & \mathbf{0} & \vdots & 1 & 0 & \vdots & 1 & 0 \\ \mathbf{0} & \mathbf{C}_{CSM}^{El} & \mathbf{0} & \mathbf{C}_{FSM}^{El} & \vdots & 0 & 1 & \vdots & 0 & 1 \end{bmatrix} \\
&\times \begin{bmatrix} \mathbf{x}_{Plant}(t_i) \\ \mathbf{x}_{disturb}(t_i) \\ \mathbf{x}_{e_I}(t_i) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{dCSM}^{Az} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{dCSM}^{El} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{dFSM}^{Az} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{dFSM}^{El} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{u}(t_i) \quad (3.60)
\end{aligned}$$

The model is then written in incremental form in terms of the perturbation states as

$$\begin{aligned}
\begin{bmatrix} \delta \mathbf{x}_{Plant}(t_{i+1}) \\ \delta \mathbf{x}_{disturb}(t_{i+1}) \\ \delta \mathbf{x}_{e_I}(t_{i+1}) \end{bmatrix} &= \begin{bmatrix} \Phi_{CSM}^{Az} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \mathbf{0} & \Phi_{CSM}^{El} & \mathbf{0} & \mathbf{0} & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \Phi_{FSM}^{Az} & \mathbf{0} & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Phi_{FSM}^{El} & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \vdots & \Phi_{disturb}^{1,1} & 0 & \vdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \vdots & 0 & \Phi_{disturb}^{2,2} & \vdots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{C}_{CSM}^{Az} & \mathbf{0} & \mathbf{C}_{FSM}^{Az} & \mathbf{0} & \vdots & 1 & 0 & \vdots & 1 & 0 \\ \mathbf{0} & \mathbf{C}_{CSM}^{El} & \mathbf{0} & \mathbf{C}_{FSM}^{El} & \vdots & 0 & 1 & \vdots & 0 & 1 \end{bmatrix} \\
&\times \begin{bmatrix} \delta \mathbf{x}_{Plant}(t_i) \\ \delta \mathbf{x}_{disturb}(t_i) \\ \delta \mathbf{x}_{e_I}(t_i) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{dCSM}^{Az} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{dCSM}^{El} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{dFSM}^{Az} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{dFSM}^{El} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \delta \mathbf{u}(t_i) \quad (3.61)
\end{aligned}$$

From Equation (3.42), the cost function is

$$\begin{aligned}
J(\delta \hat{\mathbf{x}}(t_i^+), \delta \mathbf{u}(t_i)) &= \frac{1}{2} E \left[\begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_f^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_f^+) \\ \delta \mathbf{x}_{e_I}(t_f^+) \end{bmatrix}^T \mathbf{X}_f \begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_f^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_f^+) \\ \delta \mathbf{x}_{e_I}(t_f^+) \end{bmatrix} + \right. \\
&\quad \left. + \sum_{i=0}^N \left(\begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_i^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_i^+) \\ \delta \mathbf{x}_{e_I}(t_i^+) \end{bmatrix}^T \mathbf{X} \begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_i^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_i^+) \\ \delta \mathbf{x}_{e_I}(t_i^+) \end{bmatrix} + \delta \mathbf{u}^T(t_i) \mathbf{U} \delta \mathbf{u}(t_i) \right) \right]
\end{aligned}$$

The matrices of Equation (3.61) are shown in Appendix A.3.1. In this research, the steady-state control gains are used, thus the cost function from Equation (3.43) for the steady-state control law is

$$\begin{aligned} \bar{J}(\delta \hat{\mathbf{x}}(t_i^+), \delta \mathbf{u}(t_i)) &= \frac{1}{2} E \left[\lim_{N \rightarrow \infty} \frac{1}{t_N} \sum_{i=0}^N \left(\begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_i^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_i^+) \\ \delta \mathbf{x}_{e_I}(t_i) \end{bmatrix}^T \mathbf{X} \begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_i^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_i^+) \\ \delta \mathbf{x}_{e_I}(t_i) \end{bmatrix} + \right. \right. \\ &\quad \left. \left. + \delta \mathbf{u}^T(t_i) \mathbf{U} \delta \mathbf{u}(t_i) \right) \right] \end{aligned} \quad (3.62)$$

When the LTR technique is used, the cost function (from Equation (3.44)) is

$$\begin{aligned} \bar{J}(\delta \hat{\mathbf{x}}(t_i^+), \delta \mathbf{u}(t_i)) &= \frac{1}{2} E \left[\lim_{N \rightarrow \infty} \frac{1}{t_N} \sum_{i=0}^N \left(\begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_i^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_i^+) \\ \delta \mathbf{x}_{e_I}(t_i) \end{bmatrix}^T \left(\mathbf{X} + q_{LTR} \begin{bmatrix} \mathbf{H}^T \\ \vdots \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{H} & \vdots & \mathbf{0} \end{bmatrix} \right) \right. \right. \\ &\quad \left. \left. \times \begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_i^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_i^+) \\ \delta \mathbf{x}_{e_I}(t_i) \end{bmatrix} + \delta \mathbf{u}^T(t_i) \mathbf{U} \delta \mathbf{u}(t_i) \right) \right] \end{aligned} \quad (3.63)$$

The significance of using the incremental form is that it contains no explicit pseudointegral states; therefore, initialization of the states is not required as in the original form of Equation (3.60). The steady-state PI control law for this design is

$$\begin{aligned} \mathbf{u}(t_i) &= \mathbf{u}(t_{i-1}) - \bar{\mathbf{G}}_c \begin{bmatrix} \hat{\mathbf{x}}_{Plant}(t_i^+) - \hat{\mathbf{x}}_{Plant}(t_{i-1}^+) \\ \hat{\mathbf{x}}_{disturb}(t_i^+) - \hat{\mathbf{x}}_{disturb}(t_{i-1}^+) \\ \hat{\mathbf{y}}_c(t_{i-1}^+) - \mathbf{y}_d \end{bmatrix} \\ &= \mathbf{u}(t_{i-1}) - \bar{\mathbf{G}}_c \begin{bmatrix} \hat{\mathbf{x}}_{Plant}(t_i^+) - \hat{\mathbf{x}}_{Plant}(t_{i-1}^+) \\ \hat{\mathbf{x}}_{disturb}(t_i^+) - \hat{\mathbf{x}}_{disturb}(t_{i-1}^+) \\ \hat{\mathbf{y}}_c(t_{i-1}^+) \end{bmatrix} \end{aligned} \quad (3.64)$$

since $\mathbf{y}_d = \mathbf{0}$, where $\bar{\mathbf{G}}_c$ is determined by Equation (3.30) and (3.31) using the LQG dynamics model and cost function just presented. The \mathbf{X} and \mathbf{U} matrices have been left

as variables to be tuned during simulation to determine the cost function that yields best performance, and q_{LTR} is tuned for the desired robustness.

The design has the zero-mean steady-state error property of a type-1 system as long as $\det(\bar{\mathbf{G}}_{c_2}) \neq 0$ where $\bar{\mathbf{G}}_{c_2}$ is a square matrix of the last two columns of $\bar{\mathbf{G}}_c$ associated with the two controlled output states (11). This can be shown from Equation (3.64), assuming that some steady-state output of the control system is attained:

$$\mathbf{u}_{ss} = \mathbf{u}_{ss} - \bar{\mathbf{G}}_{c_1}[\mathbf{x}_{ss} - \mathbf{x}_{ss}] - \bar{\mathbf{G}}_{c_2}\mathbf{y}_{c_{ss}} \quad (3.65)$$

where

$$\mathbf{x}_{ss} = \begin{bmatrix} \mathbf{x}_{plant} \\ \mathbf{x}_{disturb} \end{bmatrix}_{ss}$$

Equation (3.65) implies that $\bar{\mathbf{G}}_{c_2}\mathbf{y}_{c_{ss}} = \mathbf{0}$, therefore $\mathbf{y}_{c_{ss}} = \mathbf{0}$ indicating zero-mean steady-state error, where $\bar{\mathbf{G}}_{c_1}$ is a matrix of the first 29 columns of $\bar{\mathbf{G}}_c$ associated with the Kalman filter model states and the subscript ss denotes a steady-state value.

3.3.3.1 Stability. This section shows that the LQG controller designed is stable with respect to the system models of Equation (3.58), Equation (3.59), Equation (3.61), and the cost function of Equation (3.62), based on the stability criteria presented earlier in this chapter. The ranks of \mathbf{M}_C and \mathbf{M}_O with respect to the LQR (\mathbf{M}_C and \mathbf{M}_O defined by Equations (3.45) and (3.46) are

$$rank(\mathbf{M}_C) = 29 = n_{LQR} - 2$$

$$rank(\mathbf{M}_O) = 31 = n_{LQR}$$

where n_{LQR} is the number of states in the LQR model. The system described by Equations (3.61) and (3.62) is stabilizable since the only two uncontrollable states are the tilt disturbance states, which are stable modes from the first order lag shaping filter. The

system is also detectable since \mathbf{M}_O is of full rank. \mathbf{M}_O is based on the weighting matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{0} & \vdots & \mathbf{0} & \mathbf{0} \\ \cdots & \vdots & \cdots & \cdots \\ \mathbf{0} & \vdots & 10 & 0 \\ \mathbf{0} & \vdots & 0 & 10 \end{bmatrix}$$

which is used in most of the simulations. The significance of the values chosen for \mathbf{X} is discussed in the next chapter.

The ranks of \mathbf{M}_C and \mathbf{M}_O with respect to the Kalman filter (\mathbf{M}_C and \mathbf{M}_O defined by Equation (3.47) and (3.48) are

$$rank(\mathbf{M}_C) = 29 = n_{KF}$$

$$rank(\mathbf{M}_O) = 29 = n_{KF}$$

where n_{KF} is the number of states in the Kalman filter model. The system described by Equations (3.58) and (3.59) is stabilizable and detectable since \mathbf{M}_C and \mathbf{M}_O are of full rank. The stability criteria of the LQR and Kalman filter are met, therefore, the closed-loop system of the plant and LQG-designed PI controller is stable (at nominal conditions).

The LTR technique applied to the LQG controller design results in a modified \mathbf{X} matrix, depending on the q_{LTR} value used in Equation (3.44). The modified \mathbf{X} matrix could result in a different value for $rank(\mathbf{M}_O)$ with respect to the LQR, however, the system is still detectable since there are no unstable modes: the plant model poles are all in the left half of the real and imaginary plane, and the disturbance states and pseudointegral states are all stable. Therefore, the LQG/LTR designs presented in Chapter 4 are also stable for any q_{LTR} value.

3.4 Summary

This chapter presented atmospheric turbulence and LQG/LTR theory along with the details of the LQG/LTR design developed. An LQG-based PI tracker was proposed in

which the pseudointegral of the regulation error is regulated to zero. The next chapter describes the simulation environment, simulation results, and analysis of the tracker design's performance and robustness properties.

IV. Simulation and Analysis

4.1 Simulation

4.1.1 Tilt and Tip Simulation. This simulation provides a method to develop true atmosphere-induced tilt and tip values of the wavefront across the telescope aperture over a particular time interval. The simulation develops a single phase screen based on the PSD defined by Equation (3.5) at each sample time using a Fourier Series technique. This technique is based on the conventional method of developing colored noise samples from a known power spectral density, the frequency components of which can be chosen at will. Each phase screen is a discretization of the phase distortion of the two-dimensional wavefront at the telescope aperture. Each discretized phase value is associated with a two-dimensional, (x, y) , position coordinate of the aperture. Determination of the phase screen from one sample time to the next is based on the "frozen turbulence" hypothesis and a designated atmospheric wind speed (4). The hypothesis states that, with respect to an observation point, the temporal variations in atmospheric turbulence due to the dynamics of the turbulence occur much more slowly than variations resulting from motion of the atmosphere due to wind. The time scale of variations due to atmospheric motion is approximately one second, while the time scale based on dynamics is typically on the order of ten seconds. The simulation produces phase screens over a time interval of one and four seconds, making the use of the hypothesis valid. The MATLAB[®] simulation code is presented in Appendix B.1. The code was provided by the Electrical and Computer Engineering department of the Air Force Institute of Technology and modifications were made to develop time-correlated phase screens (rather than independent) and to extract the tilt and tip modes from the screens.

4.1.1.1 Short Description of Simulation Code. In general, the Fourier series representation of the phase is

$$\phi(x, y) = \sum_{p=1}^{\infty} \sum_{m=1}^{\infty} c_{p,m} e^{j2\pi(\kappa_{x_p}x + \kappa_{y_m}y)} \quad (4.1)$$

where $c_{p,m}$ is a normally distributed random variable defined by

$$E[c_{p,m}] = 0$$

$$E[|c_{p,m}|^2] = \Delta\kappa_{x_p}\Delta\kappa_{y_m}\Phi_n(\kappa_{x_p}, \kappa_{y_m}) \quad (4.2)$$

where $\Phi_n(\kappa_{x_p}, \kappa_{y_m})$ is the two-dimensional von Karmon PSD and the product, $\Delta\kappa_{x_p}\Delta\kappa_{y_m}$, is the area of a spatial frequency region (4). The one-dimensional PSD is calculated using Equation (3.5) for a range of frequencies spaced logarithmically so that the lower frequencies (containing most of the power) are discretized more finely. Let a frequency region be defined as the region between any two successive frequencies, κ_{high} and κ_{low} , where $\kappa_{high} > \kappa_{low}$. The two-dimensional PSD can be described as the result of rotating the one-dimensional PSD 360° about $\kappa = 0$. Let an annular region be defined as a frequency region rotated 360° around $\kappa = 0$. Using trapezoidal integration, the two-dimensional PSD for each annular region is determined. Within each frequency region, a frequency value κ_{mid} is iteratively chosen such that

$$\Phi_n(\kappa_{mid}) \times area_{annular\ region} = \Phi_{n_{annular\ region}}^{2D}$$

where $\Phi_n(\kappa_{mid})$ is found from Equation (3.5), $area_{annular\ region} = \pi(\kappa_{high}^2 - \kappa_{low}^2)$, and $\Phi_{n_{annular\ region}}^{2D}$ is the two-dimensional PSD for the annular region. Lastly, each annular region is divided into 32 slices of equal area. Each slice is associated with a particular angle θ_{slice} and two-dimensional PSD,

$$\Phi_{n_{slice}}^{2D}(\kappa_{mid}) = \frac{area_{annular\ region}}{32} \times \Phi_n(\kappa_{mid})$$

As just described, the two-dimensional PSD is a function of κ_{mid} and θ_{slice} , and can be represented on a cartesian coordinate system as a function of κ_x and κ_y from

$$\kappa_x = \kappa_{mid} \cos(\theta_{slice})$$

$$\kappa_y = \kappa_{mid} \sin(\theta_{slice})$$

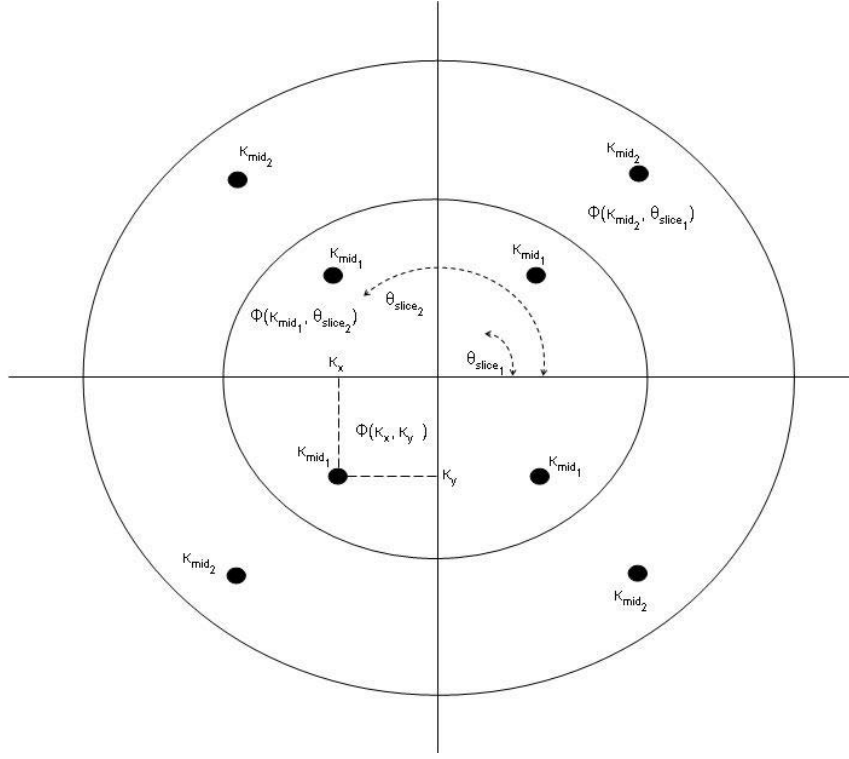


Figure 4.1 Simplified Example of Two-Dimensional PSD Coordinate Frames

Figure 4.1 depicts the coordinate frames just described. Normally distributed, zero-mean, unity-variance, random numbers are developed for each (κ_x, κ_y) and are multiplied by the corresponding $\sqrt{\Phi_{n_{slice}}(\kappa_{x_p}, \kappa_{y_m})}$. The multiplication provides the random numbers, $c_{p,m}$, with the variance of Equation (4.2) noting that $\Delta\kappa_{x_p}\Delta\kappa_{y_m}$ represents the area which is already included in $\Phi_{n_{slice}}(\kappa_{x_p}, \kappa_{y_m})$. The random numbers, frequency coordinates (κ_x, κ_y) , and discrete grid of (x, y) aperture position coordinates are used in Equation (4.1), with the infinite upper bound on n and m replaced by the finite number of frequency coordinates used, to determine the phase screen $\phi_{screen}(x, y)$. The (x, y) grid values are shifted equally in the x and y directions according to the specified wind speed to represent the atmosphere moving across the aperture. The same random numbers are used to create each phase screen, within a given time sequence, since the screens are time-correlated.

The tilt and tip modes of each of the phase screens are determined using Equations (3.10) or (3.11), (3.12), and (3.7) or (3.8). The integration in Equation (3.12) is computed as the dot product of the Zernike polynomial and the phase screen. In the simulation, the

Zernike polynomials are developed by applying a separate (x, y) coordinate grid, normalized by $r \leq 1$, to Equations (3.10) and (3.11). Since $r \leq 1$ in Equations (3.10) and (3.11) and the aperture radius used in the simulation is 1.75 meters, the Zernike polynomial grids have to be scaled by $\frac{1}{\sqrt{1.75^2\pi}}$ before performing the dot product, which is equivalent to $W(\alpha)$ in Equation (3.12). The tilt and tip Zernike coefficients are found by

$$a_{tilt/tip} = (Z_{tilt/tip_{scaled}} \cdot \phi_{screen}) \Delta x \Delta y$$

where Δx and Δy are the x and y direction spacings between the position grid elements. Equations (3.10) and (3.11) are used with the scaled Zernike polynomial grids to determine the tilt and tip phase modes. Equation (3.55) is used to calculate the angular tilt and tip. The angular tilt and tip is determined in this manner at each simulation sample time.

4.1.1.2 Simulation Parameters. Wind speeds are iteratively chosen and used in Equation (3.3) to determine the appropriate r_0 value from Equation (3.4), such that the Greenwood frequency values attained from Equation (3.13) best match the desired values listed in Table 3.1. Using this method, r_0 is dependent on the wind speed. The resulting r_0 , wind speed, and Greenwood frequency values are:

$$v = 5 \text{ m/s}, r_0 = 5.12 \text{ cm}, f_G = 41.95 \text{ Hz}$$

$$v = 10 \text{ m/s}, r_0 = 5.09 \text{ cm}, f_G = 84.43 \text{ Hz}$$

$$v = 22 \text{ m/s}, r_0 = 4.94 \text{ cm}, f_G = 191.62 \text{ Hz}$$

To simplify calculations, $r_0 = 5 \text{ cm}$ is used for each Greenwood frequency. Therefore, the values in Table 4.1 are the actual values used in the simulation. Additional simulation parameters include the aperture diameter, wavelength, and sample time shown in Table 3.1, as well as inner and outer scale values of $l_0 = 0.1 \text{ mm}$ and $L_0 = 100 \text{ m}$. Phase screen grid sizes are 129×129 and 65×65 for run times of 1 sec and 4 sec, respectively. The grid size is reduced for the 4 sec run time to keep simulation time reasonable.

Table 4.1 Tilt and Tip Simulation Parameters

$v(m/s)$	$r_0(cm)$	$f_G(Hz)$
5	5	43.0
10	5	86.0
22	5	189.2

4.1.1.3 *Simulation Results.* Table 4.2 lists the simulation scenarios and the angular tilt and tip values attained from each scenario are shown in Figures 4.2 to 4.5.

Table 4.2 Tilt and Tip Simulation Scenarios

Scenario	$f_G(Hz)$	Run Time (sec)
1	43.0	4
2	86.0	1
3	86.0	4
4	189.2	4

Comparisons of the theoretical structure function to the structure functions resulting from the simulated phase screens are shown in Figures 4.6 to 4.9. The theoretical structure function is

$$D_\phi(\rho) = 2 \frac{\rho^{5/3}}{\left(\frac{r_0}{2.1}\right)^{5/3}} \quad (4.3)$$

Equation (4.3) is found by expressing Equation (3.2) in terms of r_0 using Equation (3.4). The structure function resulting from the smaller grid size is not as accurate as the structure function from the 129×129 grid since the aperture is discretized at larger spacing intervals. The phase screen structure functions are determined by computing Equation (3.1) empirically using the simulated phase screens. Samples of $(\phi(\rho_1) - \phi(\rho_1 - \rho))^2$ in the horizontal and vertical directions from a single phase screen are averaged for each possible ρ value ($\rho_{\min} = \text{grid spacing size}$, $\rho_{\max} = d$). The averaged values from each phase screen are then averaged over all phase screens within a scenario for each ρ value. The MATLAB[®] code used to calculate the structure functions is included in Appendix B.1. Note that in Figures 4.7 to 4.9 the phase screen structure functions become less accurate at larger ρ

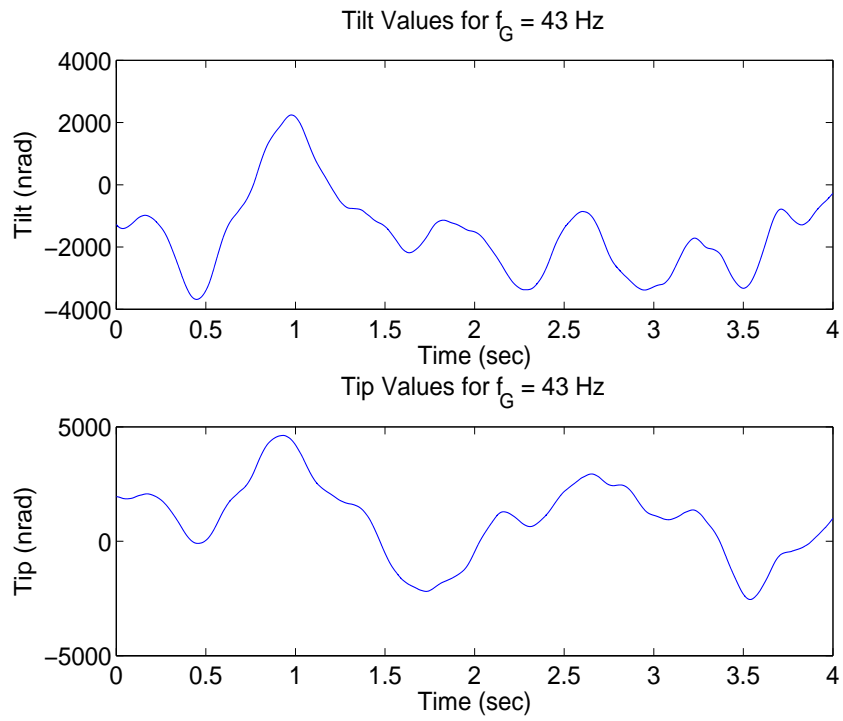


Figure 4.2 Scenario 1 Tilt and Tip Values

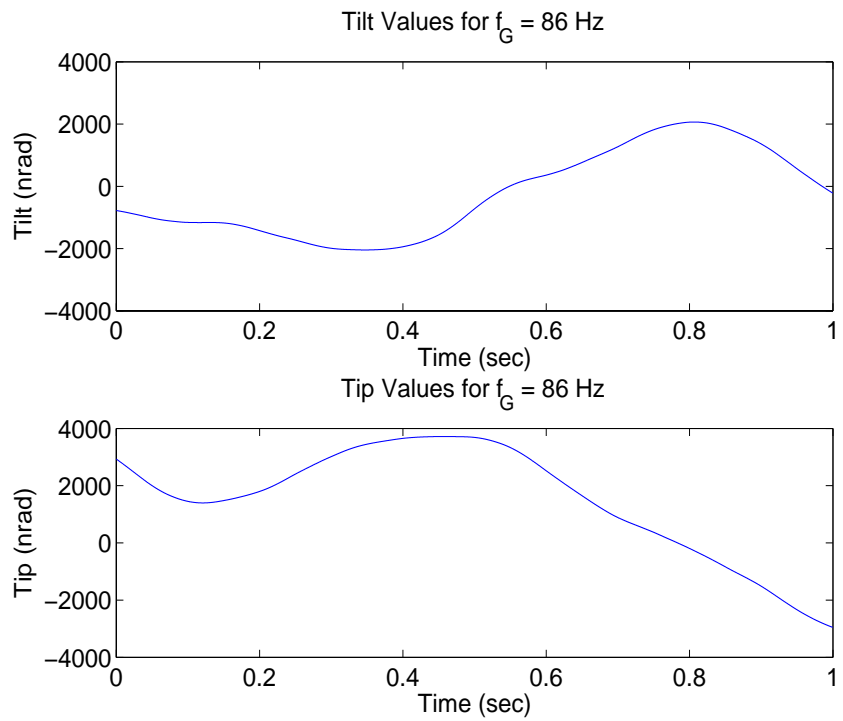


Figure 4.3 Scenario 2 Tilt and Tip Values

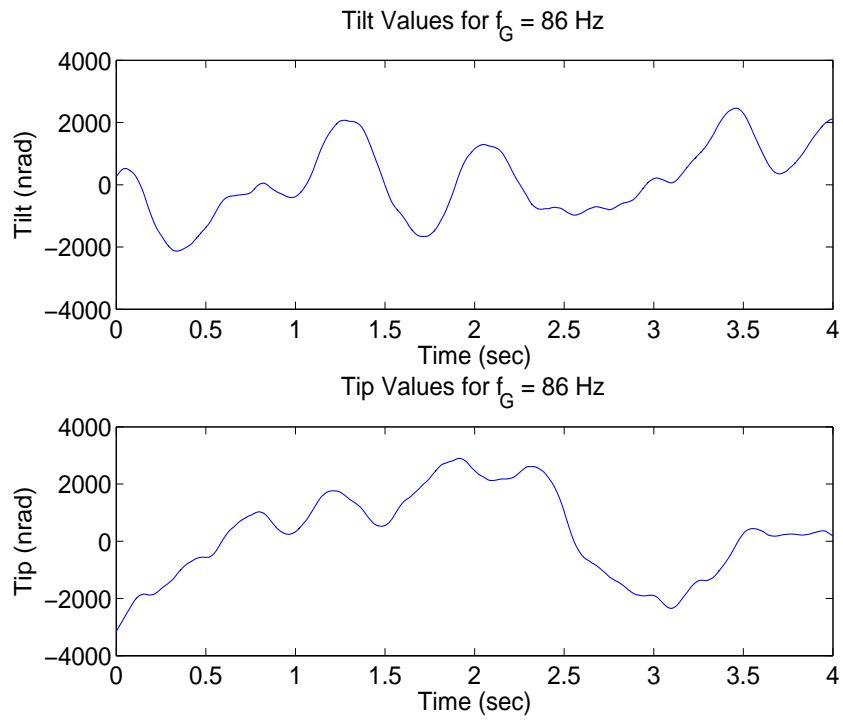


Figure 4.4 Scenario 3 Tilt and Tip Values

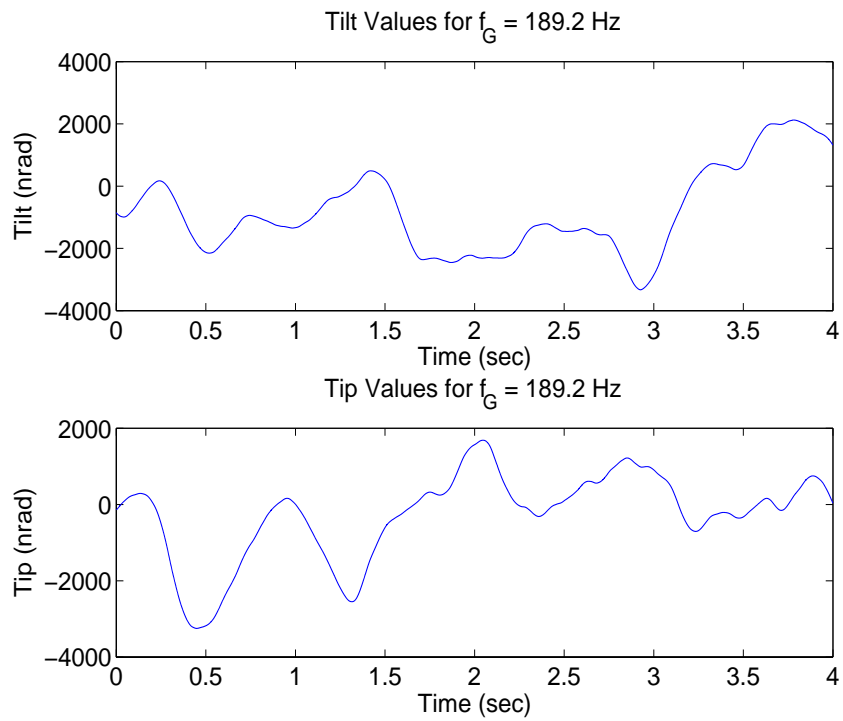


Figure 4.5 Scenario 4 Tilt and Tip Values

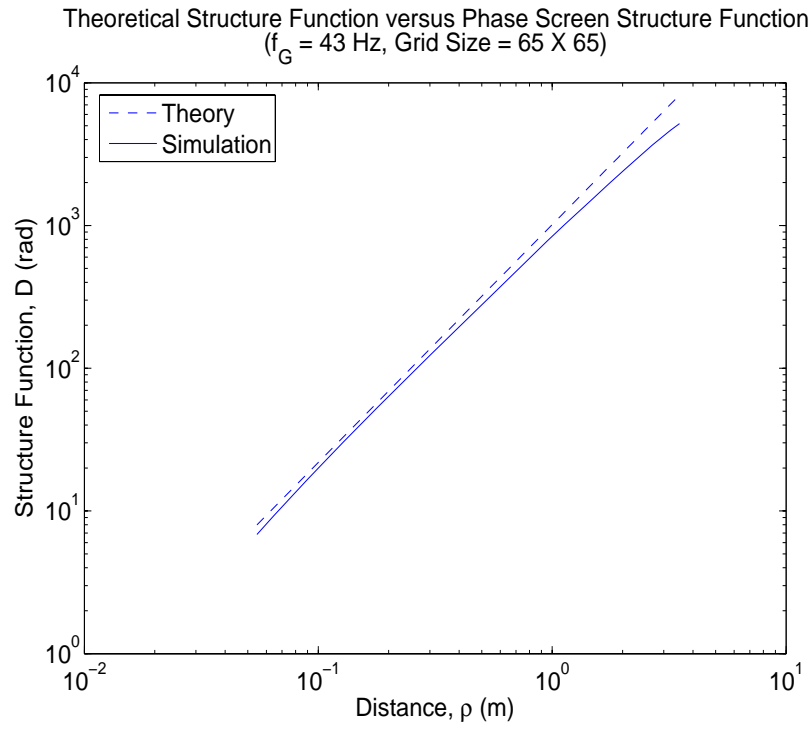


Figure 4.6 Scenario 1 Structure Function Comparison

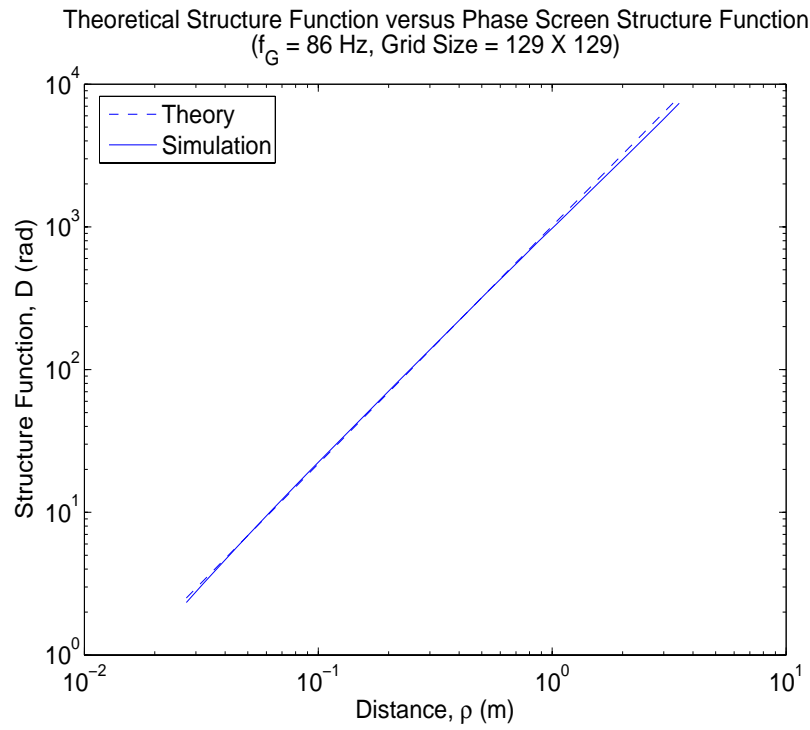


Figure 4.7 Scenario 2 Structure Function Comparison

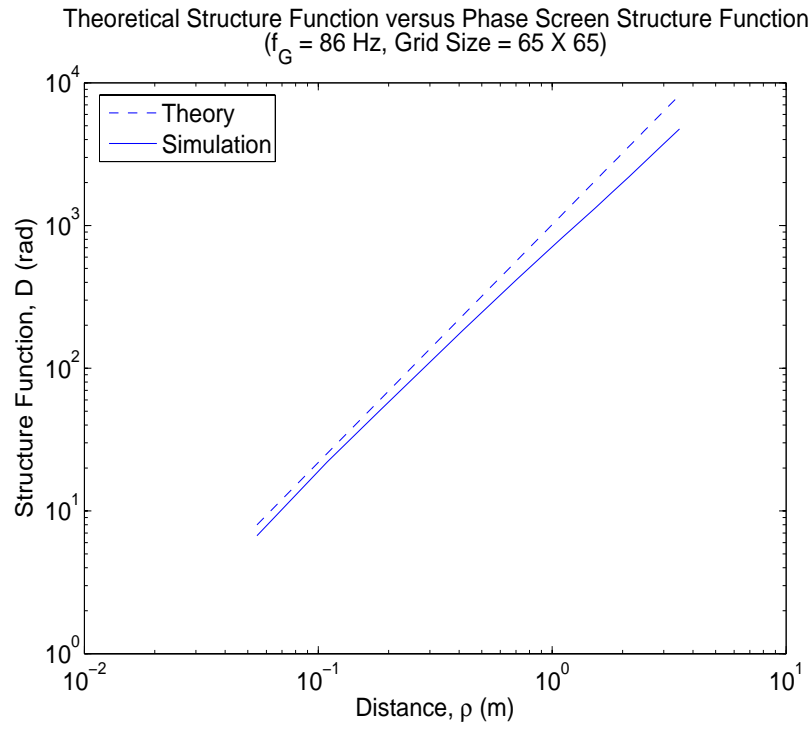


Figure 4.8 Scenario 3 Structure Function Comparison

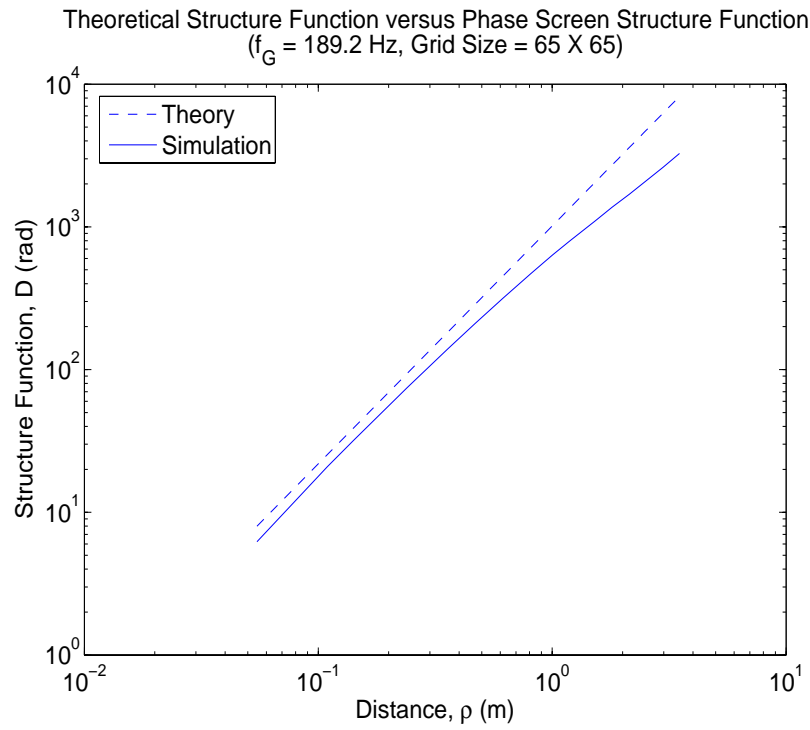


Figure 4.9 Scenario 4 Structure Function Comparison

values. This does not necessarily indicate that the simulated turbulence is less accurate at low frequencies; rather, it is probably due to the fact that there are not as many samples to average for the larger ρ values. For example, $(\phi(\rho_1) - \phi(\rho_1 - \rho))^2$ for $\rho_{\max} = d$ can only be sampled from one edge of the aperture to the other, whereas for ρ_{\min} there are numerous samples within the edges of the aperture.

4.1.2 Control Loop Simulation. This simulation provides a method of analyzing the performance of the controller design within the control loop. The simulation consists of a Simulink[®] model in conjunction with MATLAB[®] code. The MATLAB[®] code is used to compute the steady-state Kalman filter gain and optimal control gain based on the discrete-time Kalman filter and LQG models presented in Chapter 3. The gains are determined by numerically solving Equations (3.31) and (3.40). The Simulink[®] model consists of all the control loop components and simulates the dynamics of the control loop for a given time interval. The MATLAB[®] code is included in Appendix B.3.

4.1.2.1 Simulink[®] Model. The Simulink[®] model is shown in Figure 4.10. The Steering Mirrors block serves as the continuous-time truth model for the azimuth and elevation axes of the CSM and FSM and is based on the continuous dynamics model in Equations (3.52) and (3.53) from the transfer functions in Equations (3.49), (3.50), and (3.51). The 18th-order models of the FSM axes would be ideal to use as the truth model; however, simulation time became unreasonable when these models were implemented. Alternatively, the 10th-order FSM azimuth axis and 11th-order FSM elevation axis models were used (the same models used in the controller design). This raises a concern of the truth model fidelity not being high enough to depict the performance degradation due to the use of reduced-order design models. The Kalman filter and LQG/LTR blocks are based on the discrete-time Kalman filter and LQG models presented in Chapter 3. The actual tilt and tip disturbance values from the tilt and tip simulation are injected into the control loop at each sample time (0.2 msec) through the Disturbance block in Figure 4.10. Between successive sample times, the tilt and tip values are interpolated so that there is a continuous disturbance input. The simulation runs continuously using an ode45 (Dormand-Prince) differential equation solver to determine the system dynamics. Given

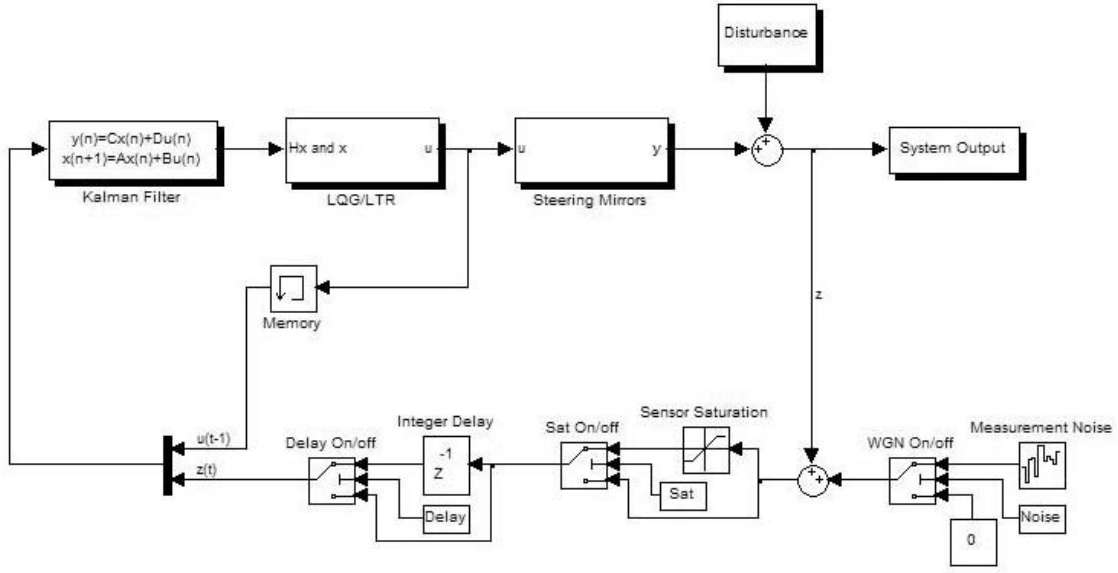


Figure 4.10 Simulink[®] Model

that the differential equation solver uses a variable step size to solve the continuous-time system numerically, it is important to specify the sampling rate of 0.2 msec s for the discrete components. The measurement of the tilt and tip disturbance and the prior control input vector are fed to the Kalman filter at the sample times only. Therefore, the optimal control inputs from the LQG/LTR block are only determined at each sample time and are held constant over each sample period, while the output of the Steering Mirrors block is solved continuously. To represent system processing time, the Integer Delay block delays the input of the measurement to the Kalman filter by a specified number of sample periods (varied throughout the simulations), resulting in the same delay of the optimal control inputs to the steering mirrors. The Measurement Noise block corrupts the tilt and tip measurements with WGN and the Saturation block represents the nonlinear characteristic of the pyramid sensor.

4.1.2.2 Pyramid Sensor Noise and Saturation Characterization.

Characterization of the pyramid sensor noise and region of nonlinearity is determined from experimental data provided by the SOR (6). The data are pyramid sensor tilt and tip output measurement values corresponding to commanded angular tilt and tip values on a received light beam. The data were collected by directing a light beam with commanded

tilt and tip at the sensor and recording the sensor output. The data were collected over a range of -2220 nrad to 2220 nrad in the azimuth axis and -1734 nrad to 1734 nrad in the elevation axis. The experiment was repeated ten times to gain ten sensor output values for every commanded value.

The pyramid sensor tilt and tip output values range from negative one to positive one according to the following quad cell algorithm:

$$tilt_{sensor} = \frac{I_1 + I_3 - (I_2 + I_4)}{\sum_{i=1}^4 I_i} \quad (4.4)$$

$$tip_{sensor} = \frac{I_3 + I_4 - (I_1 + I_2)}{\sum_{j=1}^4 I_j} \quad (4.5)$$

where I is the measured intensity of the beam spot on the sensor. Figure 4.11 aids in understanding Equations (4.4) and (4.5). For each commanded tilt and tip, the mean

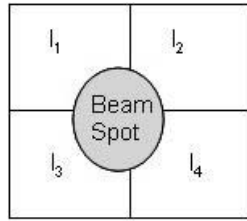


Figure 4.11 Quad Cell Depiction

pyramid sensor output was determined and a linear fit was made within the linear region of the curve of mean values. This is shown in Figures 4.12 and 4.13 along with all of the pyramid sensor output data samples (note the different horizontal axis scales in the two figures). The sensor exhibits a region of near-linearity beyond which the sensor saturates and the mean curve becomes nonlinear. The output values of the quad cell sensor that fall within the linear region can be converted to nanoradians by dividing by the slope of the linear fit line shown in Figures 4.12 and 4.13. To characterize the noise of the pyramid sensor, it is assumed to be zero-mean WGN with variance determined by calculating the variance of the output data samples about the mean curve. Thus, the variance for each

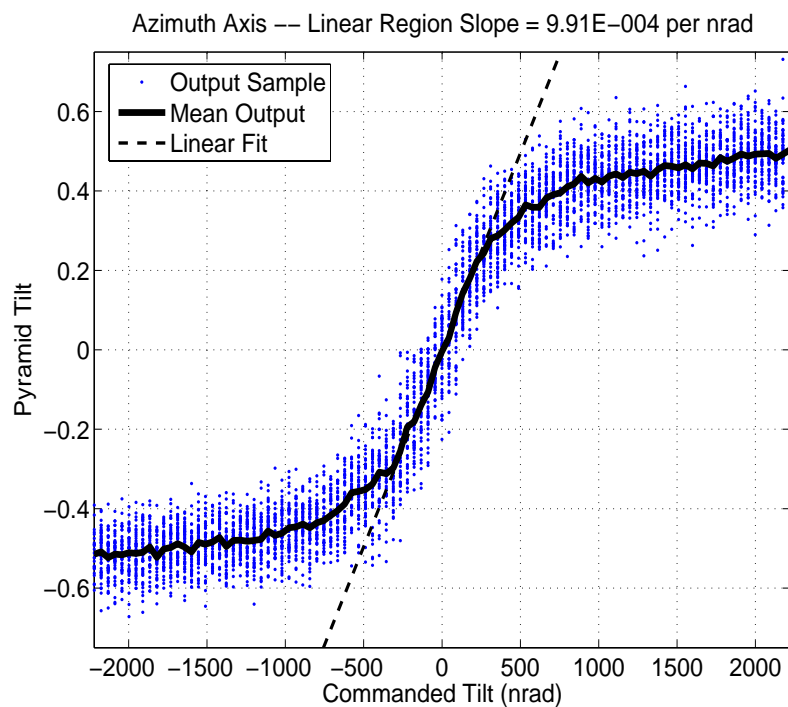


Figure 4.12 Pyramid Sensor Azimuth Axis Output Characteristics

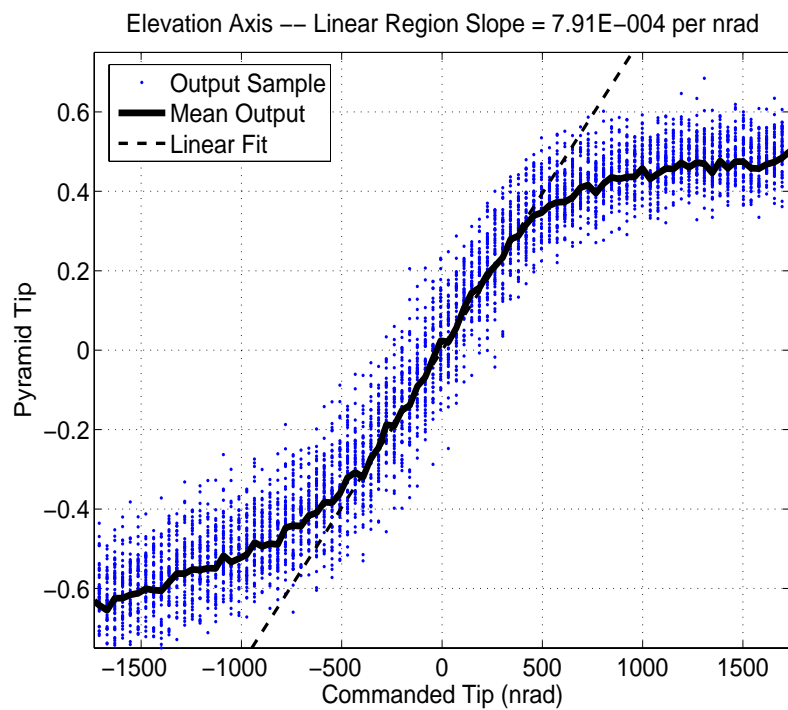


Figure 4.13 Pyramid Sensor Elevation Axis Output Characteristics

axis is calculated as

$$var_{sensor\ noise} = \frac{1}{slope^2} \left(\frac{1}{n} \sum_{i=1}^n var_{outputs_i} \right) \quad (4.6)$$

where n is the number of commanded tilt or tip values within the linear region, $slope$ is the slope of the linear fit line, and

$$var_{outputs_i} = \frac{1}{m} \sum_{j=1}^m (sample_j - mean_i)^2 \quad (4.7)$$

where $m = 10$ (the number of output samples for each commanded value), $sample_j$ is the j^{th} output sample for a given commanded value, and $mean_i$ is the average of the output samples for a given commanded value. The sensor noise variances are listed in Table 4.3 and are used as the WGN variance values in the Measurement Noise block of the Simulink[®] model. Equation (4.7) provides a biased variance of the empirical noise data

Table 4.3 Sensor Noise Variance

Axis	Sensor Noise Variance (nrad ²)
Azimuth	7412.9
Elevation	13550

rather than the preferred unbiased variance. Therefore, the values in Table 4.3 are biased and erroneously used in the simulation. However, using the equation for the unbiased variance,

$$var_{outputs_i} = \frac{1}{m-1} \sum_{j=1}^m (sample_j - mean_i)^2$$

along with Equation (4.6) produces very similar results as shown in Table 4.4 . To demonstrate the minimal impact of using the Table 4.4 values has on the final results, Chapter 4 compares outcomes from additional simulation runs using the unbiased values to the identical runs using the biased values. The saturation values used in the Simulink[®] model

Table 4.4 Unbiased Variance of Sensor Noise

Axis	Sensor Noise Variance (nrad ²)
Azimuth	7564.1
Elevation	13827

were approximated visually from Figures 4.12 and 4.13 as

$$sat_{Az} = \frac{\pm 0.30}{9.91E - 4} \approx \pm 303 \text{ nrad} \quad (4.8)$$

$$sat_{El} = \frac{\pm 0.35}{7.91E - 4} \approx \pm 443 \text{ nrad} \quad (4.9)$$

where $9.91E - 4$ is the slope of the azimuth linear fit line and $7.91E - 4$ is the slope of the elevation linear fit line. The values of ± 0.30 and ± 0.35 were chosen as the approximate minimum and maximum mean sensor output values of the linear regions. The MATLAB[®] code used to develop the sensor noise variances and the plots shown in Figures 4.12 and 4.13 was provided by the SOR and modified as needed (6). The code is shown in Appendix B.2.

4.1.2.3 Simulation Cases. Tables 4.5 and 4.6 outline the different simulation cases conducted to analyze the controller's performance. Case (27) in Table 4.5 is a baseline case simulated after analysis of the noise cases. Within each case, the elements of **Q**, **R**, **X**, and **U** are tuned in search of the best performance possible from the controller. Notice that **Q** is from the continuous-time Kalman filter model, therefore, **Q** is tuned and then discretized. In cases that incorporate the LTR technique, q_{LTR} is tuned to achieve the desired robustness. In most simulation cases, f_G is 86 Hz since it is the average Greenwood frequency. Baseline cases are those that do not incorporate the pyramid sensor noise in order to gain performance results under the most nominal conditions. Most of the cases with the delay turned on consist of multiple simulations using several different delay times. Delay times are chosen to test the control system under extreme conditions; therefore, the larger delays are not necessarily representative of the actual control loop. In Case (1), the tilt disturbance values are also used as the tip disturbance values to compare

Table 4.5 Baseline Simulation Cases (Acq = Acquisition, Sat = Saturation)

Case	Run Time (sec)	f_G (Hz)	Pyramid Sensor Noise	Acq	Tilt _{disturb} = Tip _{disturb}	Delay	Sat	LTR
1	1	86	No	No	No	No	No	No
2	1	86	No	Yes	No	No	No	No
3	1	86	No	No	Yes	No	No	No
4	1	86	No	Yes	Yes	Yes	No	No
5	1	86	No	No	No	No	Yes	No
6	1	86	No	Yes	No	No	Yes	No
7	1	86	No	No	No	Yes	No	No
8	1	86	No	Yes	No	Yes	No	No
9	1	86	No	No	No	Yes	Yes	No
10	1	86	No	Yes	No	Yes	Yes	No
11	1	86	No	No	No	Yes	Yes	Yes
12	1	86	No	Yes	No	Yes	Yes	Yes
13	1	86	No	No	No	No	No	Yes
14	1	86	No	Yes	No	No	No	Yes
27	1	86	No	No	No	Yes	No	No

Table 4.6 Sensor Noise Simulation Cases (Acq = Acquisition, Sat = Saturation)

Case	Run Time (sec)	f_G (Hz)	Pyramid Sensor Noise	Acq	Tilt _{disturb} = Tip _{disturb}	Delay	Sat	LTR
15	1	86	Yes	No	No	No	No	No
16	1	86	Yes	Yes	No	No	No	No
17	1	86	Yes	No	No	Yes	Yes	No
18	1	86	Yes	Yes	No	Yes	Yes	No
19	1	86	Yes	No	No	Yes	Yes	Yes
20	1	86	Yes	Yes	No	Yes	Yes	Yes
21	1	86	Yes	No	No	No	No	Yes
22	1	86	Yes	Yes	No	No	No	Yes
23	4	86	Yes	No	No	Yes	Yes	Yes
24	4	189.2	Yes	No	No	Yes	Yes	Yes
25	4	43	Yes	No	No	Yes	Yes	Yes
26	1	86	Yes	No	No	No	No	No

the performance of the azimuth and elevation axes subjected to identical disturbances. In non-acquisition cases, all of the tilt and tip disturbance values are equally shifted so that the values at the first sample time are zero. This enables evaluation of the steady-state system as if good tracking has already been established and prevents the need for large control inputs to the steering mirrors. *The results from the non-acquisition cases serve as the primary focus of this research in order to validate the following assumption stated in Chapter 1.*

Assumption *It is assumed that the dynamics of the steering mirrors do not change based on the size of the received inputs, although in reality the dynamics are very nonlinear in that they differ with small versus large inputs. The dynamics corresponding to small inputs are used in this research for both the controller models and to represent the mirrors in simulation (6).*

All cases involving the sensor noise consist of ten Monte Carlo runs. Figures 4.14 to 4.17 show the shifted versions of Figures 4.2 to 4.5.

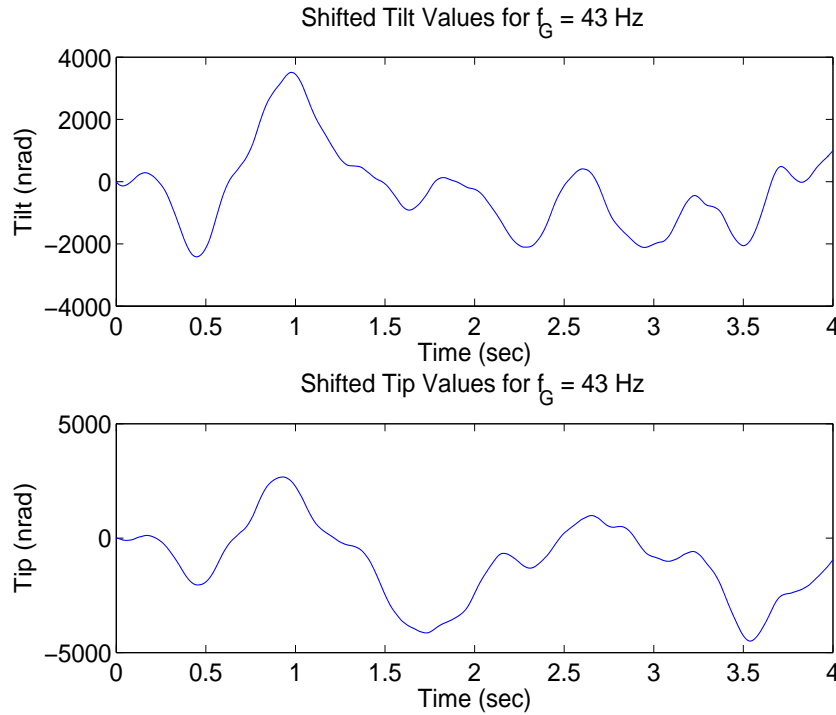


Figure 4.14 Scenario 1 Shifted Tilt and Tip Values

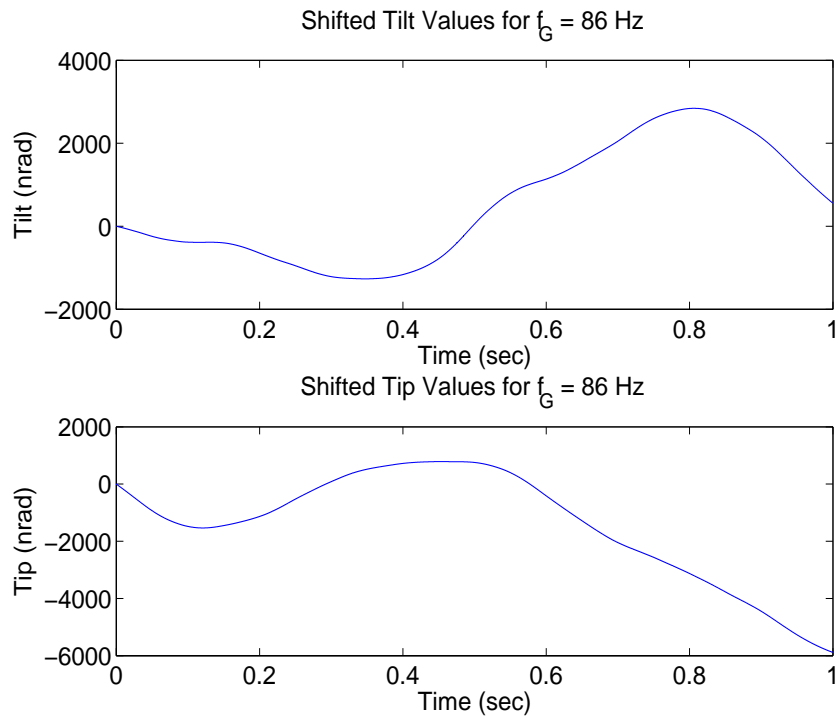


Figure 4.15 Scenario 2 Shifted Tilt and Tip Values

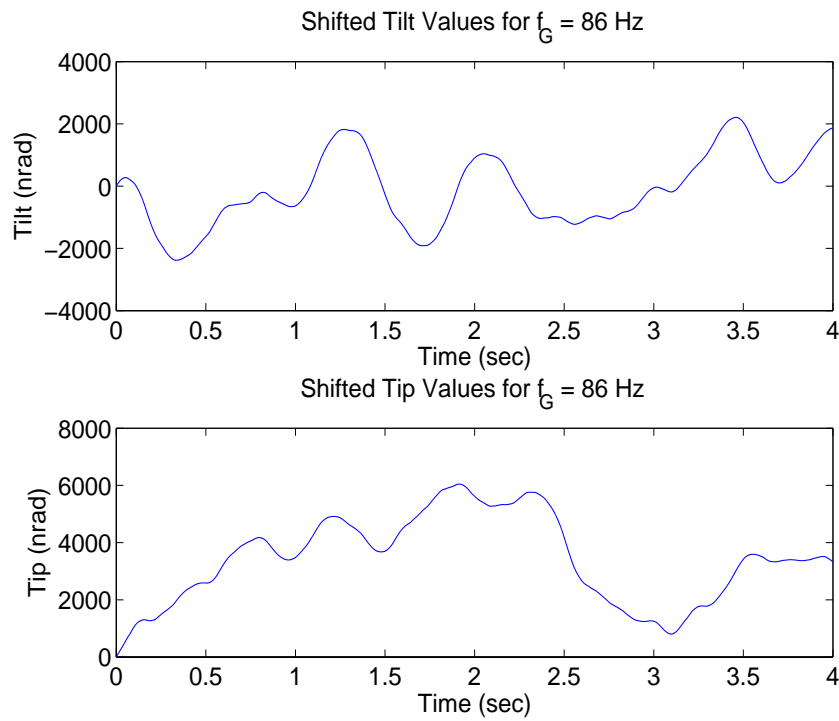


Figure 4.16 Scenario 3 Shifted Tilt and Tip Values

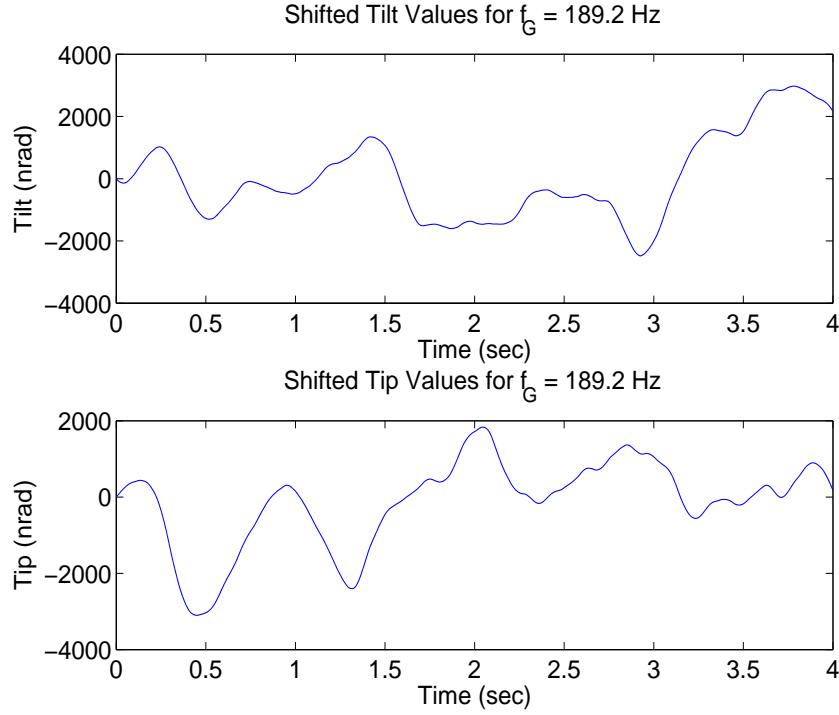


Figure 4.17 Scenario 4 Shifted Tilt and Tip Values

4.2 Analysis of Simulation Results

This section presents useful data from the simulation cases that are analyzed and discussed to show the performance and robustness properties of the system based on different tuning values and cost functions in the controller design. The primary data used to show the performance and robustness properties are the final tilt and tip values of the wavefront of light after passing through the controlled steering mirrors. This corresponds to the Final Output of the system as shown in Figure 4.10. As mentioned earlier, the ultimate goal of the control system is to drive the final tilt and tip values to zero for all time. The statistics used to evaluate controller performance are based on the *absolute value of the final tilt and tip values* so that negative and positive errors are treated the same. In the baseline simulation cases, the System Output figures display a plot of the true tilt and tip data (—) versus time, as well as the mean absolute tilt, mean absolute tip, and standard deviation values. The absolute true tilt and tip values are averaged over time to

compute the mean absolute tilt and tip as

$$\bar{\phi}_t = \frac{1}{n} \sum_{i=1}^n |\phi_i|$$

where $\bar{\phi}_t$ is the mean tilt or tip over time, ϕ_i is the true tilt or tip data, and n is the total number of tilt or tip data points collected over the simulation run. The true tilt and tip values are recorded at each integration step time of the differential equation solver since the values result from the sum of the continuous disturbance values and continuous output of the steering mirrors. The variance of the true tilt and tip values is

$$\sigma_t^2 = \frac{1}{n-1} \sum_{i=1}^n (|\phi_i| - \bar{\phi}_t)^2$$

This variance is unbiased and centered about the mean tilt or tip. The unbiased standard deviation is found from the variance as

$$\sigma_t = \sqrt{\sigma_t^2}$$

In simulation cases that incorporate the pyramid sensor noise, the System Output figures display a plot of the mean absolute true tilt and tip data (—) versus time, a plot of the standard deviation (\cdots) above and below the mean absolute true tilt and tip values versus time, as well as the overall mean absolute tilt/tip and overall standard deviation values. The mean absolute true tilt and tip values are calculated at each data collection sample time as the average over the ten Monte Carlo runs. The true tilt and tip in these cases is collected every 0.02 msec, which is one tenth of the control loop sample time. The mean absolute true tilt and tip is calculated as

$$\bar{\phi} = \frac{1}{N} \sum_{i=1}^N |\phi_i| \tag{4.10}$$

where N is the total number of Monte Carlo runs. The unbiased standard deviation about the mean absolute tilt and tip is

$$\sigma = \left(\frac{1}{N-1} \sum_{i=1}^N (|\phi_i| - \bar{\phi})^2 \right)^{\frac{1}{2}} \quad (4.11)$$

and is plotted as $\bar{\phi} \pm \sigma$. The overall mean absolute tilt and tip is the average of the mean absolute tilt and tip values (from Equation (4.10)) over time, and the overall standard deviation is the average of the standard deviation values (from Equation (4.11)) over time. The overall mean absolute tilt and tip and overall standard deviation are described by

$$\bar{\phi}_t = \frac{1}{n} \sum_{i=1}^n \bar{\phi}_i$$

$$\sigma_t = \frac{1}{n} \sum_{i=1}^n \sigma_i$$

where $\bar{\phi}$ is from Equation (4.10), σ is from Equation (4.11), and n is the total number of mean absolute tilt and tip values. Additional figures are shown and described in particular cases as needed to support the analysis.

4.2.1 Baseline Cases. All baseline cases use the actual tilt and tip disturbance values shown in Figures 4.3 and 4.15. The following sections describe each baseline case and display the control loop simulation results.

4.2.1.1 Cases (1) and (2). Most of the design parameter tuning is based on these two cases. Recall, the LQG cost function is

$$\begin{aligned} \bar{J}(\delta \hat{\mathbf{x}}(t_i^+), \delta \mathbf{u}(t_i)) &= \frac{1}{2} E \left[\lim_{N \rightarrow \infty} \frac{1}{t_N} \sum_{i=0}^N \left(\begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_i^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_i^+) \\ \delta \mathbf{x}_{e_I}(t_i) \end{bmatrix}^T \mathbf{X} \begin{bmatrix} \delta \hat{\mathbf{x}}_{Plant}(t_i^+) \\ \delta \hat{\mathbf{x}}_{disturb}(t_i^+) \\ \delta \mathbf{x}_{e_I}(t_i^+) \end{bmatrix} \right. \right. \\ &\quad \left. \left. + \delta \mathbf{u}^T(t_i) \mathbf{U} \delta \mathbf{u}(t_i) \right) \right] \end{aligned}$$

The cost function weighting matrices, \mathbf{X} and \mathbf{U} , were first tuned with the off-diagonal entries of \mathbf{Q} and \mathbf{R} as zero and with the following diagonal entries:

$$\mathbf{Q}_{diag} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 2.7535E7 & 2.7535E7 \end{bmatrix}$$

$$\mathbf{R}_{diag} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

where the last two entries of \mathbf{Q}_{diag} are from the disturbance model. The small values of \mathbf{Q}_{diag} were chosen since the Kalman filter dynamics model matches the truth model, and the small values of \mathbf{R}_{diag} were chosen since the pyramid sensor noise is not present on the measurements. Three different pairs of \mathbf{X} and \mathbf{U} matrices are compared here based on performance observed from numerous tuning scenarios. The three pairs formed three different cost functions that resulted in the best observed control system performance. The three pairs of \mathbf{X} and \mathbf{U} matrices identified as subcases (a), (b), and (c) for each case are

(a)	$\mathbf{X}_{diag} =$	$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 10 & 10 \end{bmatrix}$	$\mathbf{U}_{diag} =$	$\begin{bmatrix} 1E0 & 1E0 & 1E4 & 1E4 \end{bmatrix}$
(b)	$\mathbf{X}_{diag} =$	$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 10 & 10 \end{bmatrix}$	$\mathbf{U}_{diag} =$	$\begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$
(c)	$\mathbf{X}_{diag} =$	$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 10 & 10 \end{bmatrix}$	$\mathbf{U}_{diag} =$	$\begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$

where the off-diagonal values are zero. During the tuning process, it was discovered that changing the nonzero \mathbf{X}_{diag} values by any order of magnitude and the \mathbf{U}_{diag} values by the same order of magnitude produces identical performance results as produced by \mathbf{X}_{diag} and \mathbf{U}_{diag} prior to the magnitude changes. Simply changing just the zero values of \mathbf{X}_{diag} had either no effect or a detrimental effect on the performance. Therefore, the same \mathbf{X}_{diag} values are used in each case, while the \mathbf{U}_{diag} values are changed to modify performance. The values of \mathbf{X}_{diag} were chosen to accomplish the goal of driving the final tilt and tip values to zero. The zero values in \mathbf{X}_{diag} indicate that the plant and disturbance states are not required to be regulated to zero to minimize the cost function, while the nonzero values cause the regulation of the pseudointegral states to zero. The plant states are driven to values such that the output of the steering mirrors negates the disturbance, thus driving the pseudointegral states to zero. The \mathbf{Q}_{diag} and \mathbf{R}_{diag} values were also tuned for each cost function. The following trend was observed for each cost function: \mathbf{Q}_{diag} (other than

the two disturbance values) and \mathbf{R}_{diag} values less than those shown previously have no effect on performance, while values greater than those worsen performance. Changing the disturbance \mathbf{Q}_{diag} values from the previous values degrades the performance.

Figures 4.18 to 4.20 show the non-acquisition final tilt and tip values and Figures 4.21 to 4.23 show the acquisition final tilt and tip values. Case (1a) shows the best performance with the lowest mean tilt/tip and standard deviation and least initial transient amplitude and time. Comparing Case (1b) to Case (1c), the initial transient times are very similar while the amplitude is less in Case (1c). The smaller transient amplitude of Case (1c) leads to a smaller standard deviation than Case (1b), but Case (1c) has greater mean tilt/tip than Case (1b). Cases (2a) through (2c) show the same trend in the transient, but a trend reversal in mean tilt/tip and standard deviation. Case (2c) produces the best mean and standard deviation values. It is important to note that the mean and standard deviation values in Case (2) are much greater than those in Case (1) due to the large influence of the transient amplitude from larger initial tilt and tip disturbance values. Most importantly, Cases (2a) through (2c) do show that the system is able to regulate very large disturbances. Figures 4.24 to 4.26 display the optimal control inputs provided to the steering mirrors from the controller for non-acquisition runs. Figures 4.27 to 4.29 display the optimal control inputs for acquisition runs. The control input data are collected at the sample time of the control system. In Cases (1) and (2), the amount of control input to the CSM versus the FSM is decreased when observing subcases (a) through (c) in order. In Cases (1a) and (2a), nearly all of the control input is to the CSM, while in Cases (1c) and (2c) the control input to the CSM is almost equal to the input to the FSM. This trend is appropriate since the first two values of \mathbf{U}_{diag} , the control weighting on the CSM axes, are much less than the FSM weights in Cases (1a) and (2a), and are equal to the FSM weights in Cases (1c) and (2c). Notice that the transient trend in Cases (2a) through (2c) is consistent with the final tilt and tip data in that Case (2a) shows much larger initial inputs than Cases (2b) and (2c). Case (2c) displays more oscillation in the transient inputs than Case (2b). Throughout Cases (1) and (2), the statistics for *tip* are greater than those for *tilt*. Differences in the performance between the two axes is understandable since the tilt and tip disturbances are not the same over the time interval. Therefore, the weaker

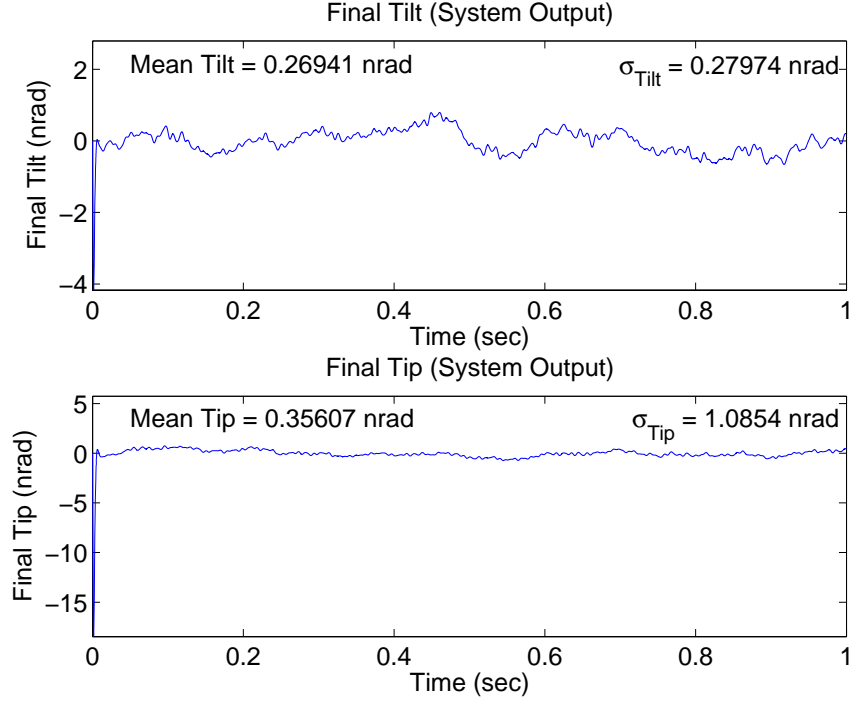


Figure 4.18 Case (1a): $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

performance in the elevation axis seen in Cases (1) and (2) cannot necessarily be attributed directly to the control design.

4.2.1.2 Cases (3) and (4). The purpose of Cases (3) and (4) is to evaluate the performance in both tilt and tip based on equal disturbance inputs, to enable a direct comparison of the axes. The tilt disturbance in Figures 4.3 and 4.15 are used as both the tilt and tip disturbances in the simulation. The simulated control system is based on the same three designs used in Cases (1) and (2). Figures 4.30 to 4.32 show the non-acquisition final tilt and tip values and Figures 4.33 to 4.35 show the acquisition final tilt and tip values. The data show that the mean tilt and mean tip values differ by as little as 0.001 nrad (in Case (3a)) and no more than 0.27 nrad (in Case (4c)). The standard deviation values differ by as little as 0.001 nrad (in Case (3a)) and no more than 1.19 nrad (in Case (4b)). Based on the data presented, the elevation axis performance is consistently better than the azimuth performance by the range of differences stated, which is not an overall significant difference. The mean and standard deviation values in Cases (3) and (4)

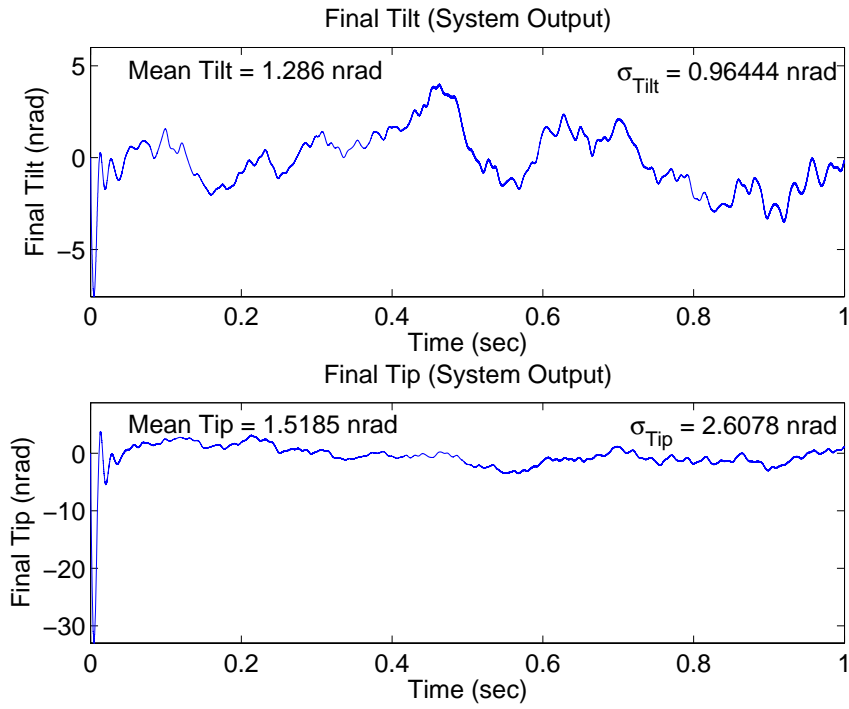


Figure 4.19 Case (1b): $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

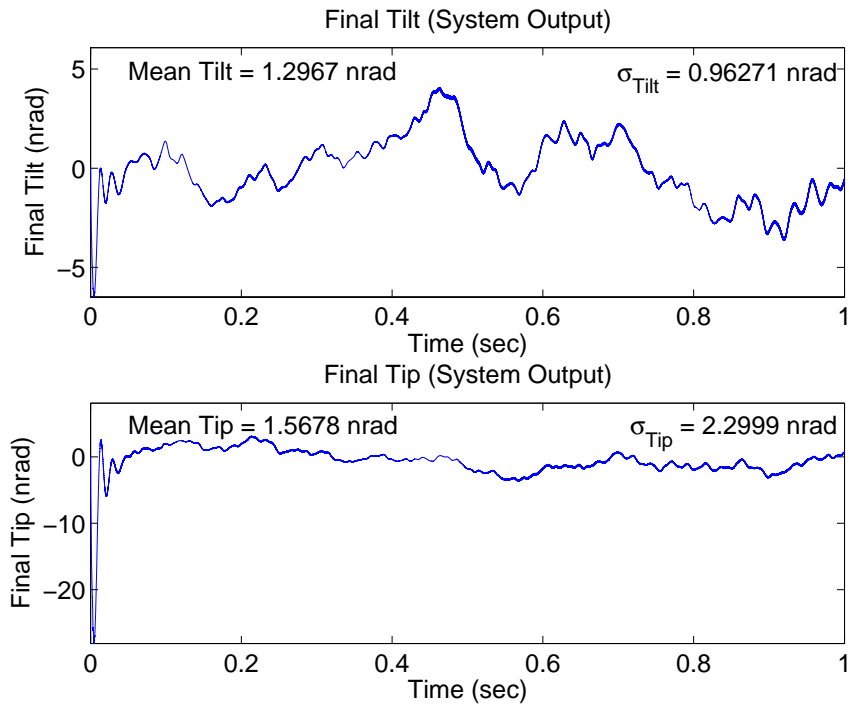


Figure 4.20 Case (1c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

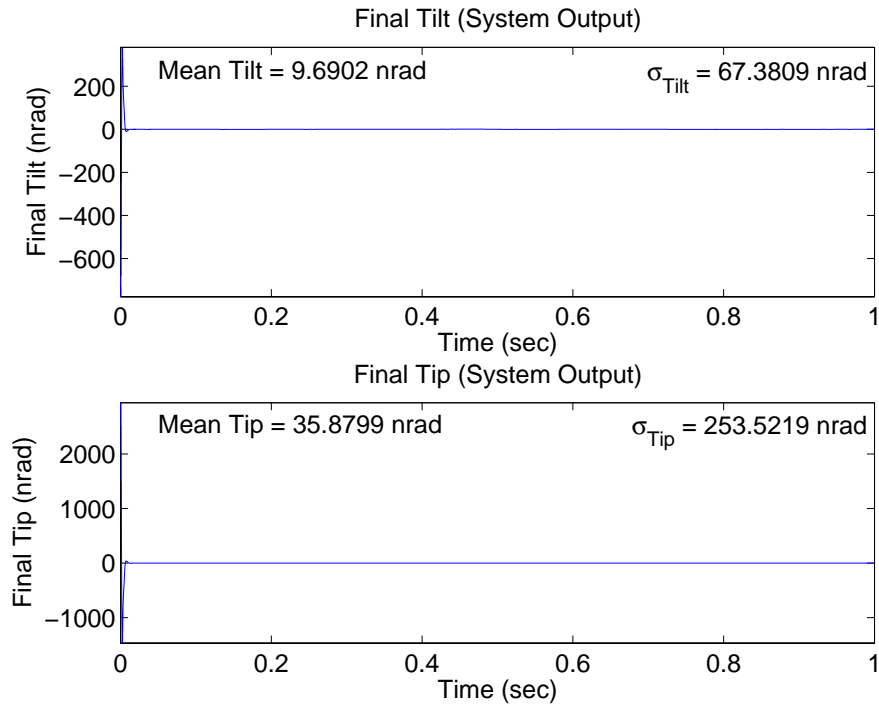


Figure 4.21 Case (2a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

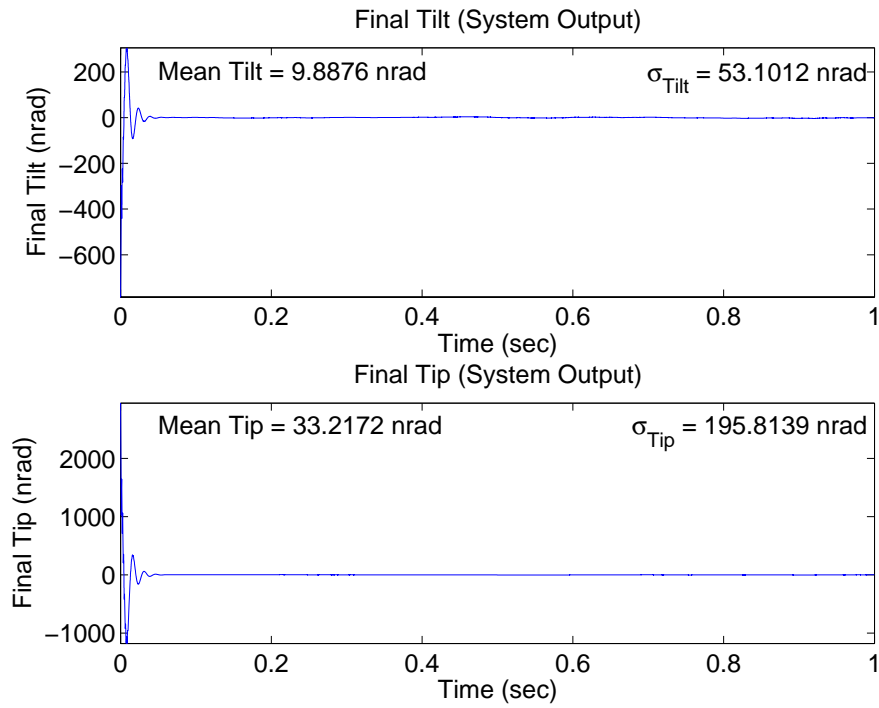


Figure 4.22 Case (2b): $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

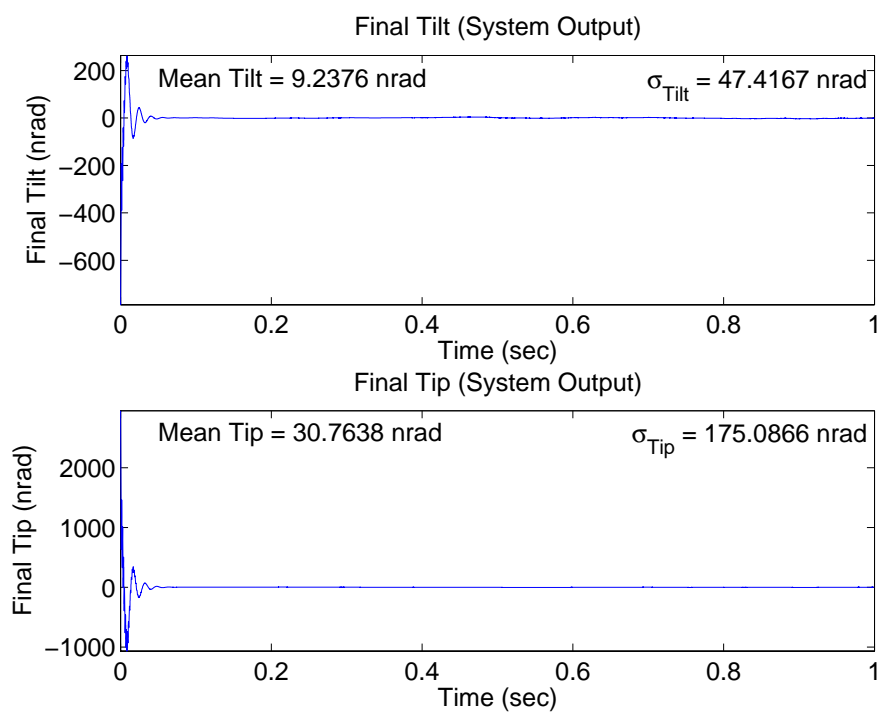


Figure 4.23 Case (2c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

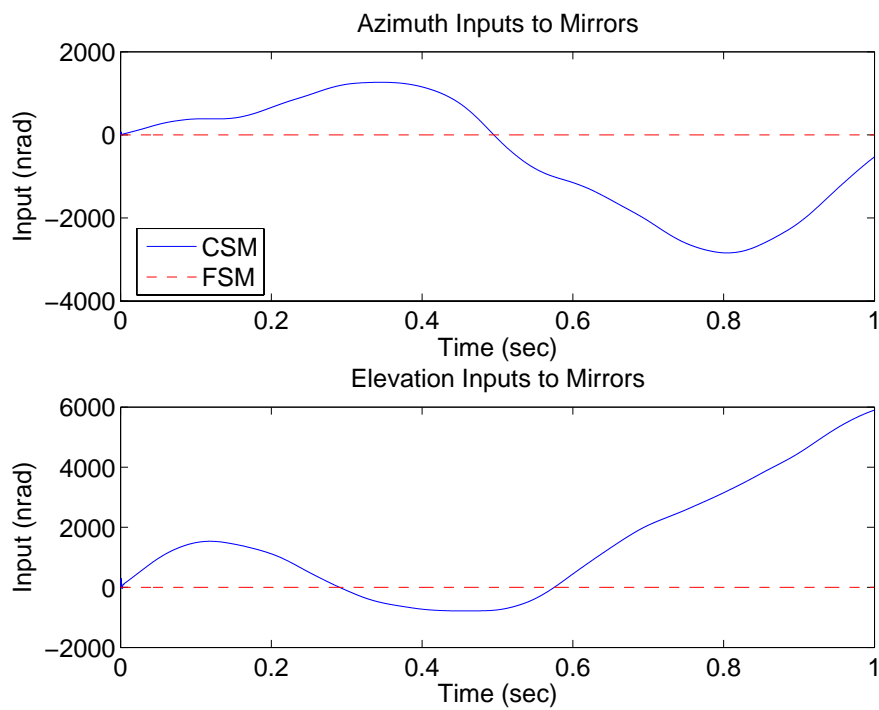


Figure 4.24 Case (1a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

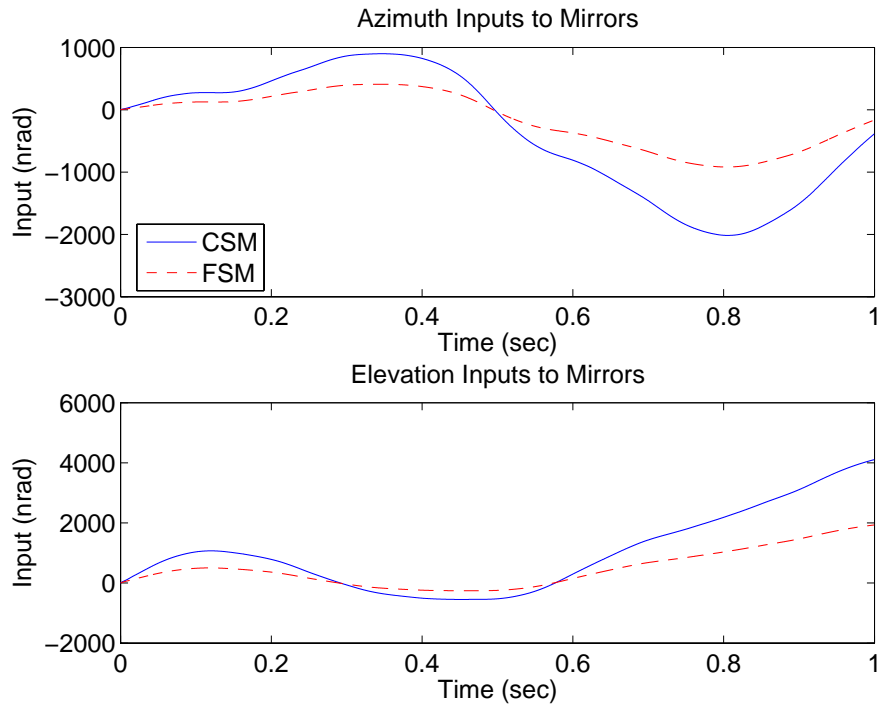


Figure 4.25 Case (1b): $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

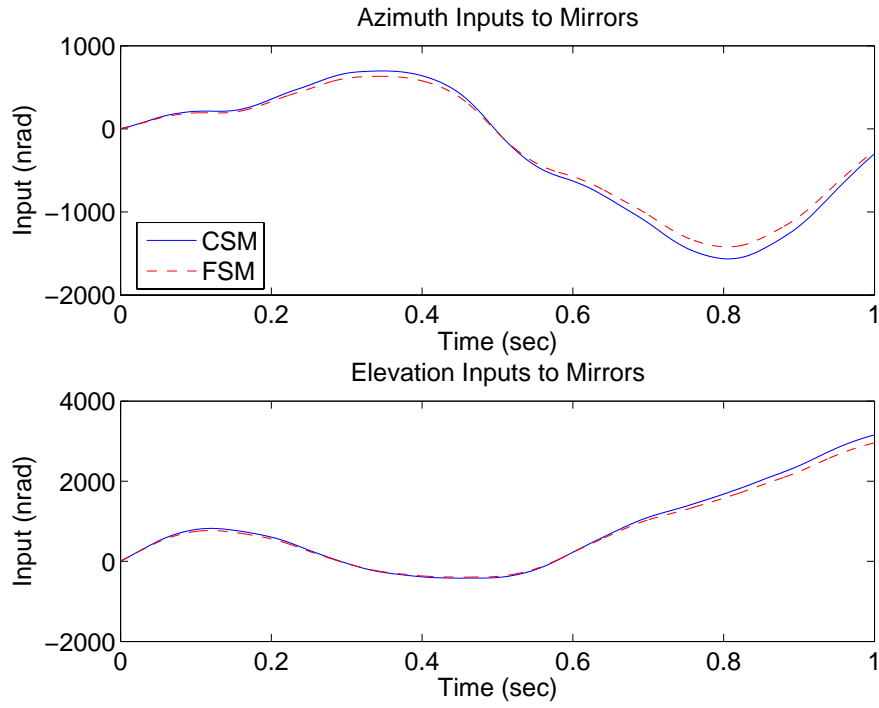


Figure 4.26 Case (1c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

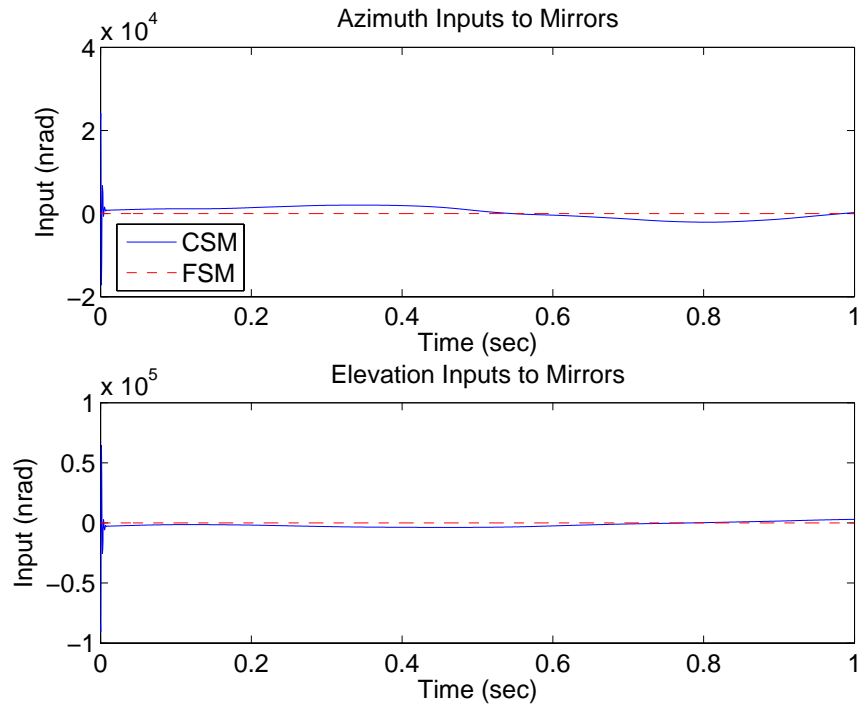


Figure 4.27 Case (2a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

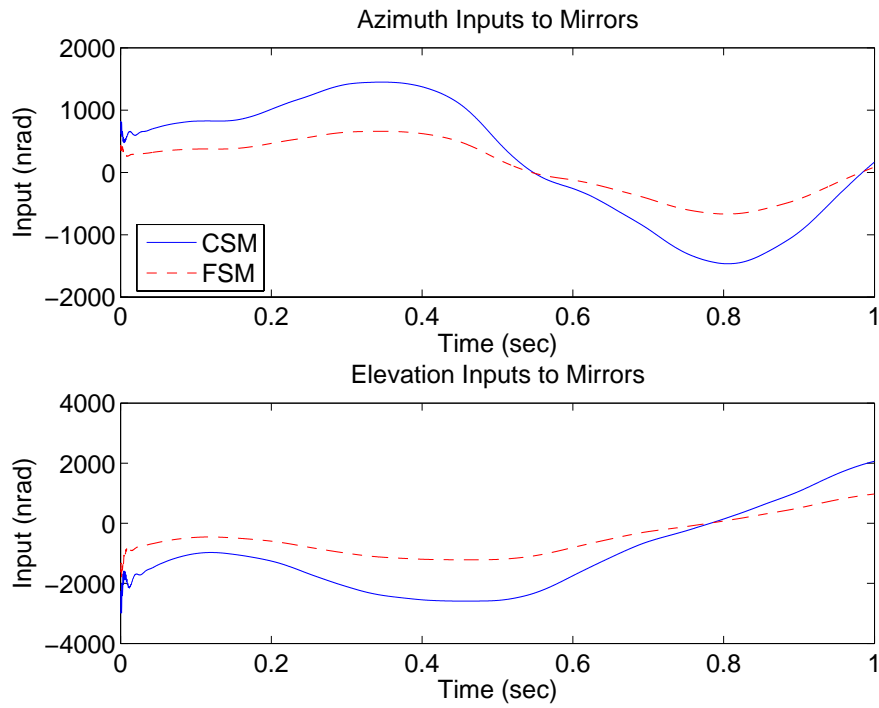


Figure 4.28 Case (2b): $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

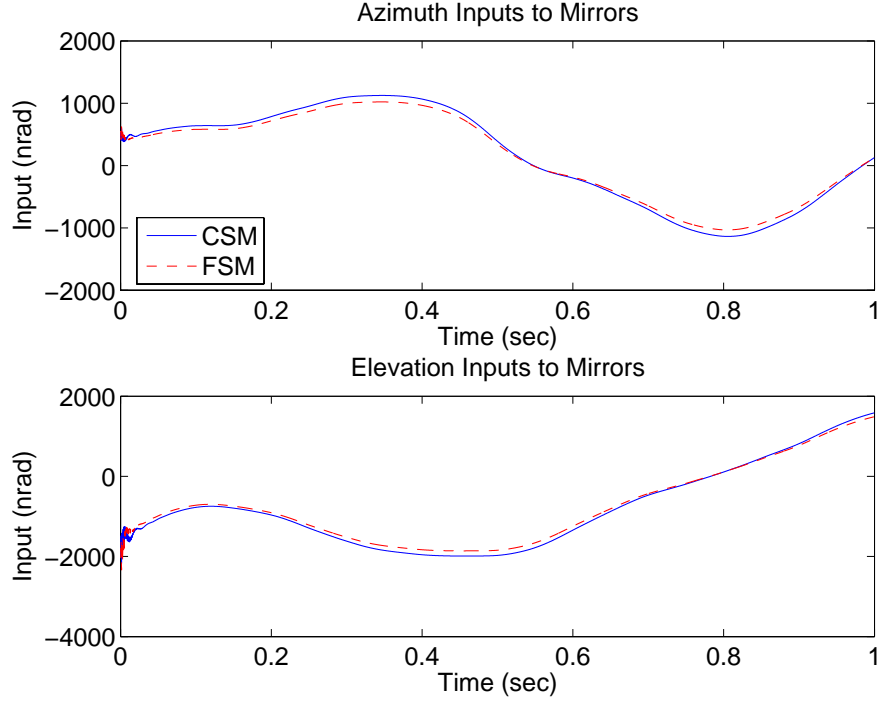


Figure 4.29 Case (2c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

are unexpectedly different from the mean tilt and standard deviation values in Cases (1) and (2). This is due to the simulation's differential equation solver using slightly different time steps which results in a slightly different interpolation of the disturbance. The plots of the final tilt and tip in Figures 4.30 to 4.35 look visually identical to Figures 4.18 to 4.23.

4.2.1.3 Cases (5) and (6). The purpose of Cases (5) and (6) is to evaluate the performance of the control system using saturated measurements from the pyramid sensor based on the values in Equations (4.8) and (4.9). The results from Case (5) are identical to Case (1), and therefore, are not shown. The results are identical since the tilt and tip disturbance starts at zero and the control system keeps the final tilt and tip well below the saturation values, so that none of the measurements during the simulation are affected by the saturation levels. Figures 4.36 to 4.38 show the mean tilt/tip and standard deviation values for Case (6). In Case (6), the mean tilt values actually improve, but by no more than 1.2 nrad (Case (6c)). The tilt standard deviation values improve by

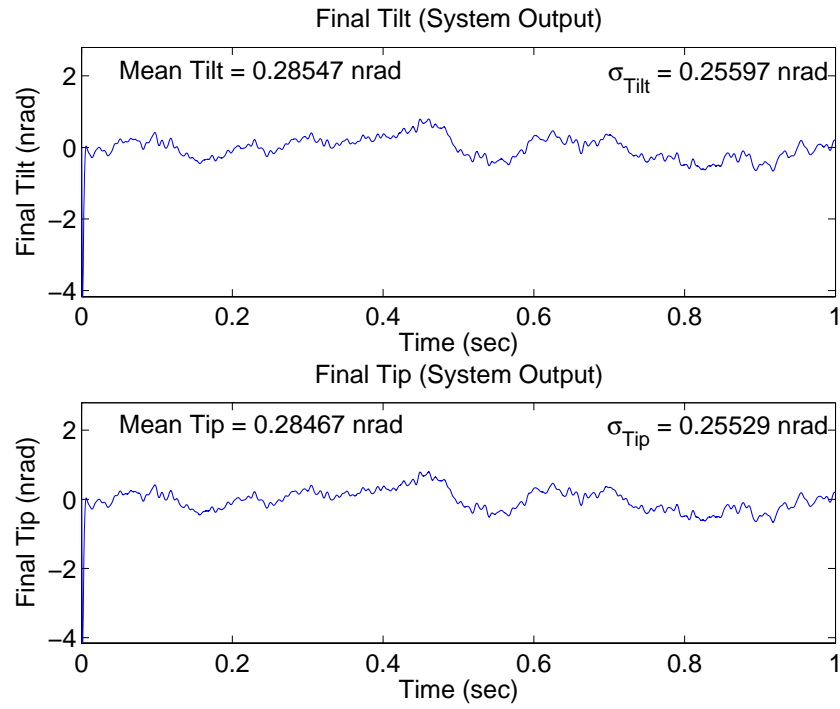


Figure 4.30 Case (3a): $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

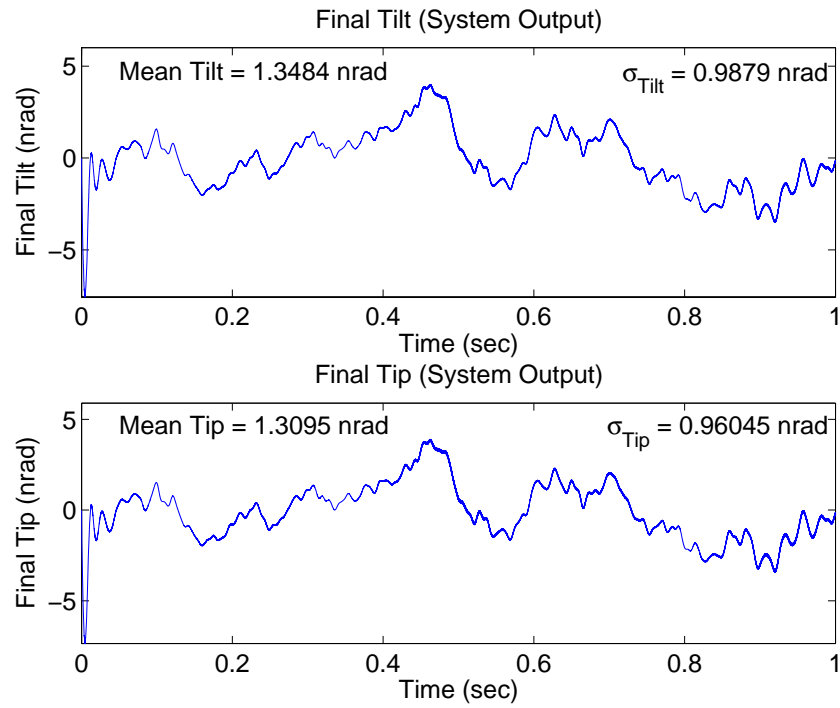


Figure 4.31 Case (3b): $\mathbf{U}_{diag} = [3E5 \quad 3E5 \quad 1E4 \quad 1E4]$

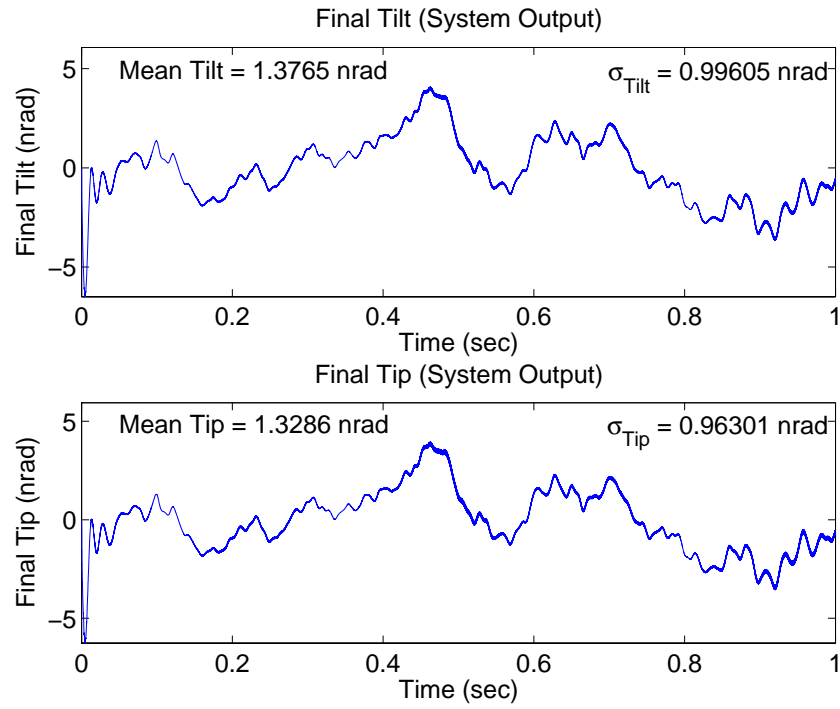


Figure 4.32 Case (3c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

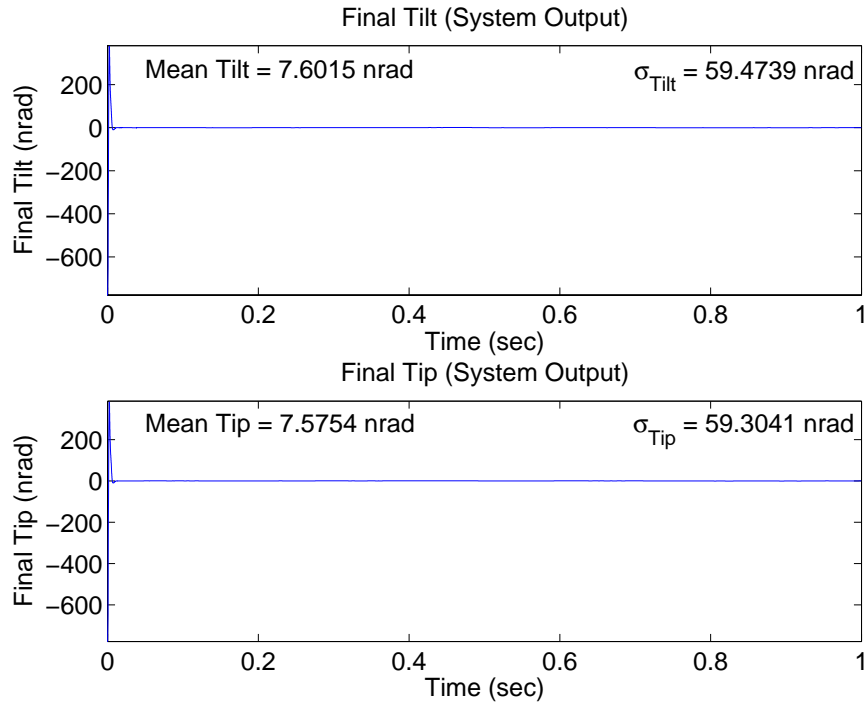


Figure 4.33 Case (4a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

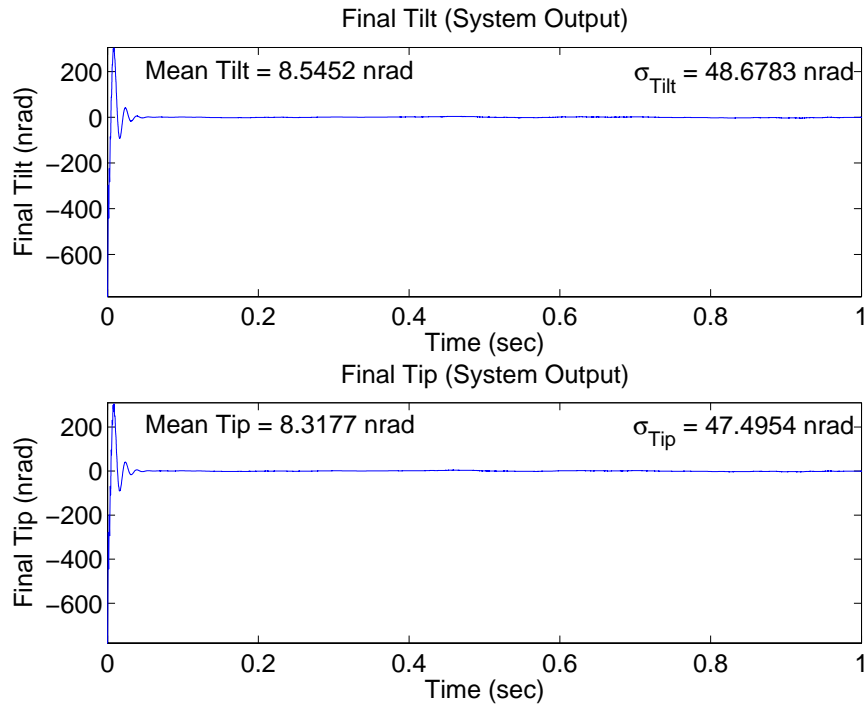


Figure 4.34 Case (4b): $\mathbf{U}_{diag} = [3E5 \ 3E5 \ 1E4 \ 1E4]$

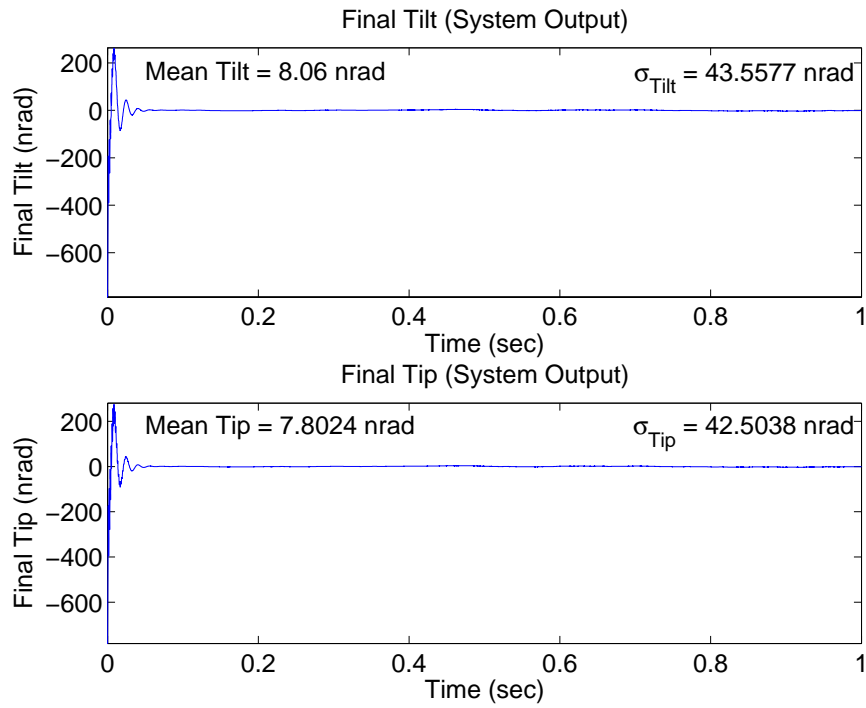


Figure 4.35 Case (4c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

no more than 4.45 nrad in Cases (6a) and (6c), and degrade by about 0.5 nrad in Case (6b). The mean tip values, however, worsen by 7 (Case (6c)) to 17 nrad (Case (6a)), and the tip standard deviation values increase by 45 (Case (6c)) to 67 nrad (Case (6b)). In Case (6), both the tilt and tip initial measurement values are saturated due to the initial disturbance values, but the final tilt and tip values are driven close to zero by the control system almost as quickly as in Case (2). The reason the azimuth axis out-performs the elevation axis is due to the larger initial tip disturbance values as shown in Figure 4.3. It appears that the initial tilt disturbance values are not large enough for the saturation to lengthen the transient time in the azimuth axis as it did in the elevation axis. In fact, the tilt saturation reduced the azimuth transient amplitude, the probable cause of the performance improvement, especially in Case (6a). Figures 4.39 to 4.41 show changes in the transient input values from Case (2), especially in the elevation axis. Case (6a) shows more oscillatory behavior for a longer period of time, while the initial inputs in Cases (6b) and (6c) seem to have reduced oscillation. The inputs beyond the transient time are identical to Case (2).

4.2.1.4 Cases (7) and (8).

One-Sample-Time Delay. The purpose of Cases (7) and (8) is to evaluate the performance and robustness of the control system to various system delay times. Cases (7.1) and (8.1) simulate the control loop with a one-sample time delay. For Case (7.1), Figures 4.42 to 4.44 show the non-acquisition final tilt and tip values, while Figures 4.45 to 4.47 show the acquisition final tilt and tip values for Case (8.1). There are no significant performance changes in Case (7.1) compared to Case (1), only a very slight degradation in mean and standard deviation values on the order of 0.01 nrad. As compared to Case (2), Case (8.1a) begins to show a degradation in mean and standard deviation values while Cases (8.1b) and (8.1c) show an unexpected improvement in performance (Case (8.1c) shows the most improvement) due to slightly reduced transient amplitudes. The control input plots are not shown since there are no significant changes from Case (1) and (2). The only noticeable change in control inputs is less transient oscillation in Case (8.1).

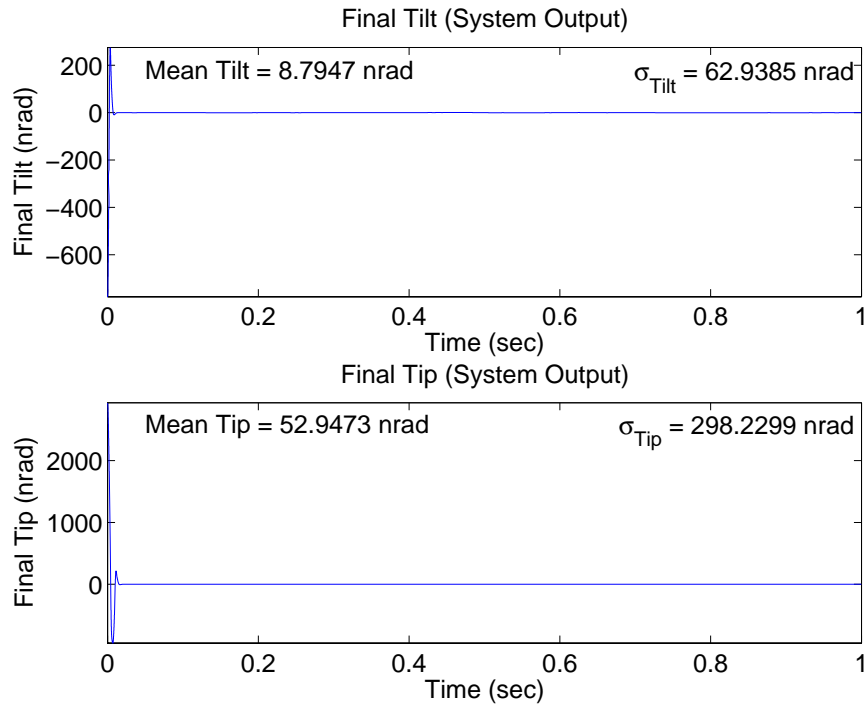


Figure 4.36 Case (6a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

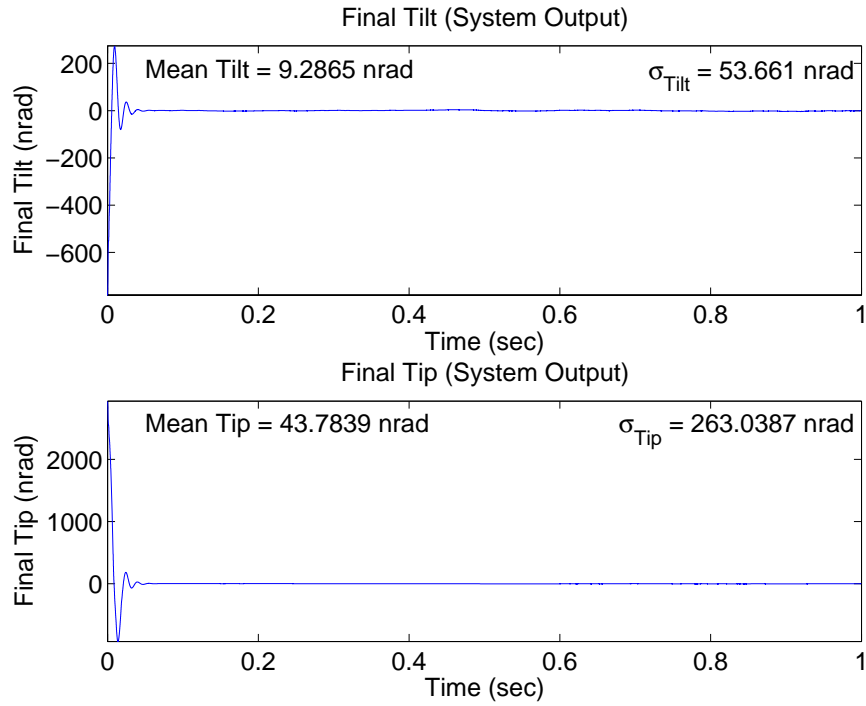


Figure 4.37 Case (6b): $\mathbf{U}_{diag} = [3E5 \ 3E5 \ 1E4 \ 1E4]$

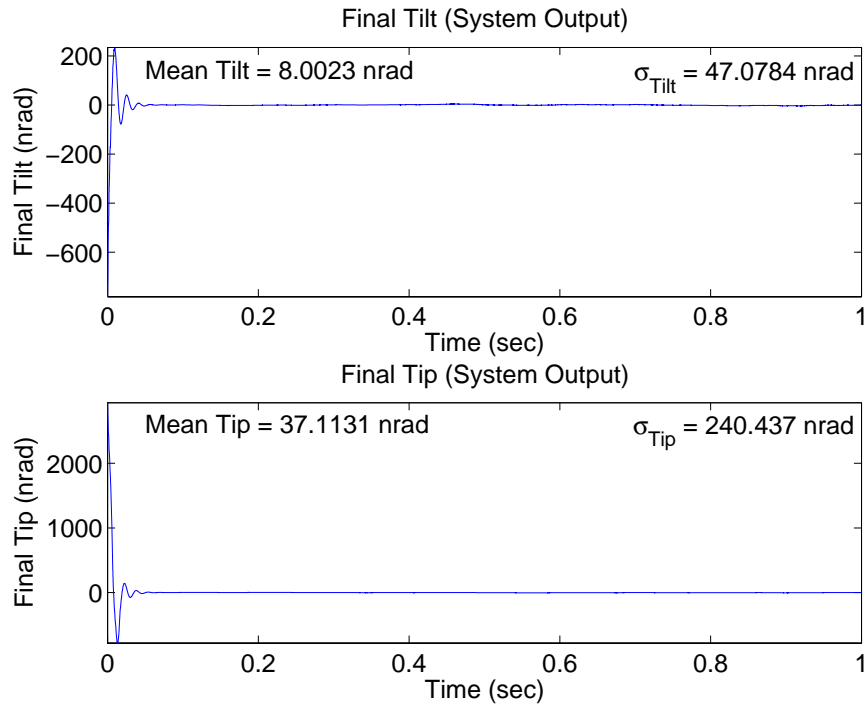


Figure 4.38 Case (6c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

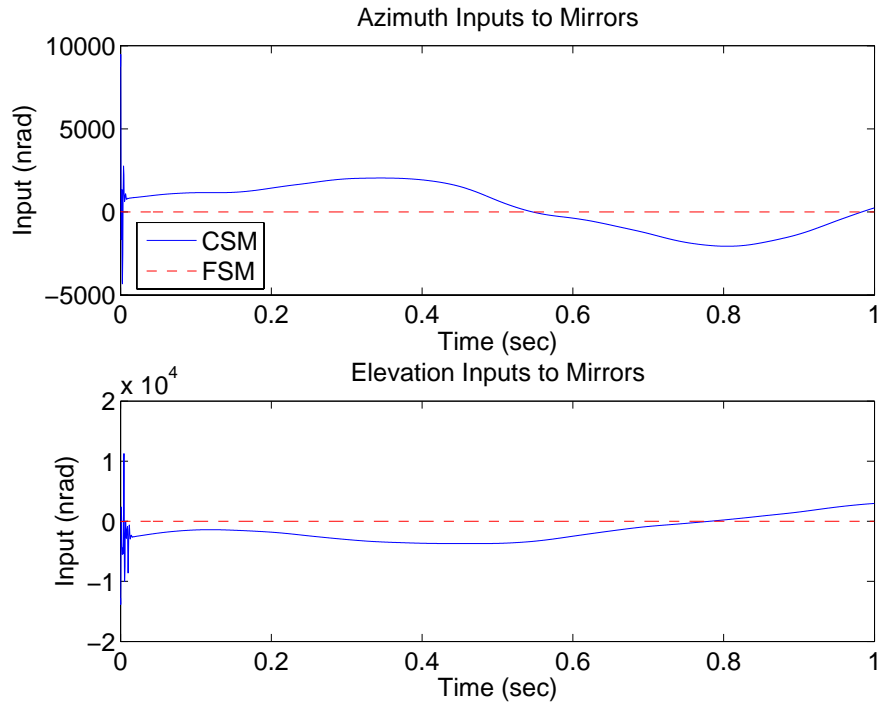


Figure 4.39 Case (6a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

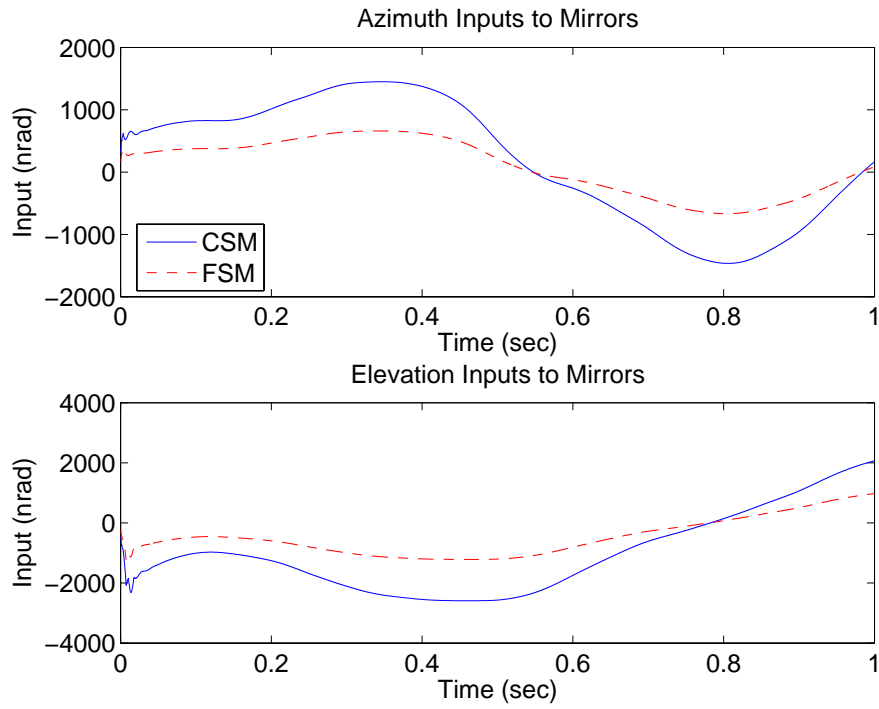


Figure 4.40 Case (6b): $\mathbf{U}_{diag} = [3E5 \ 3E5 \ 1E4 \ 1E4]$

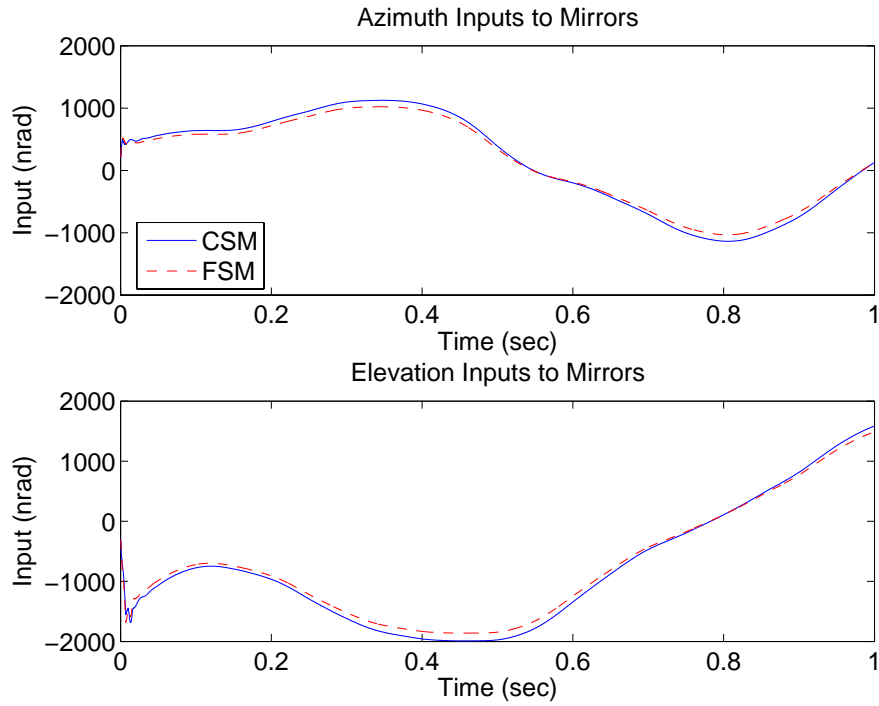


Figure 4.41 Case (6c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

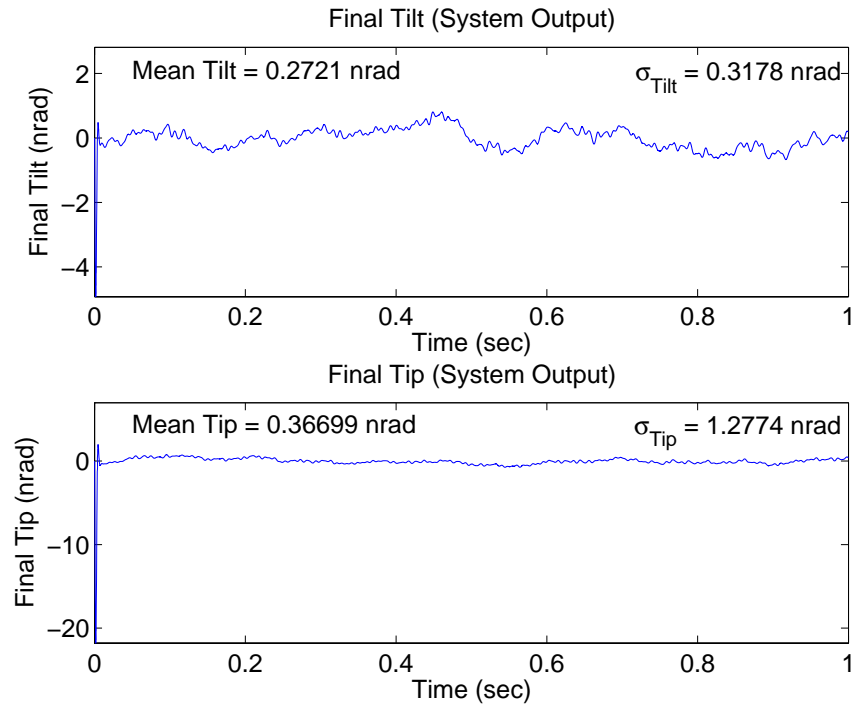


Figure 4.42 Case (7.1a): Delay 1, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

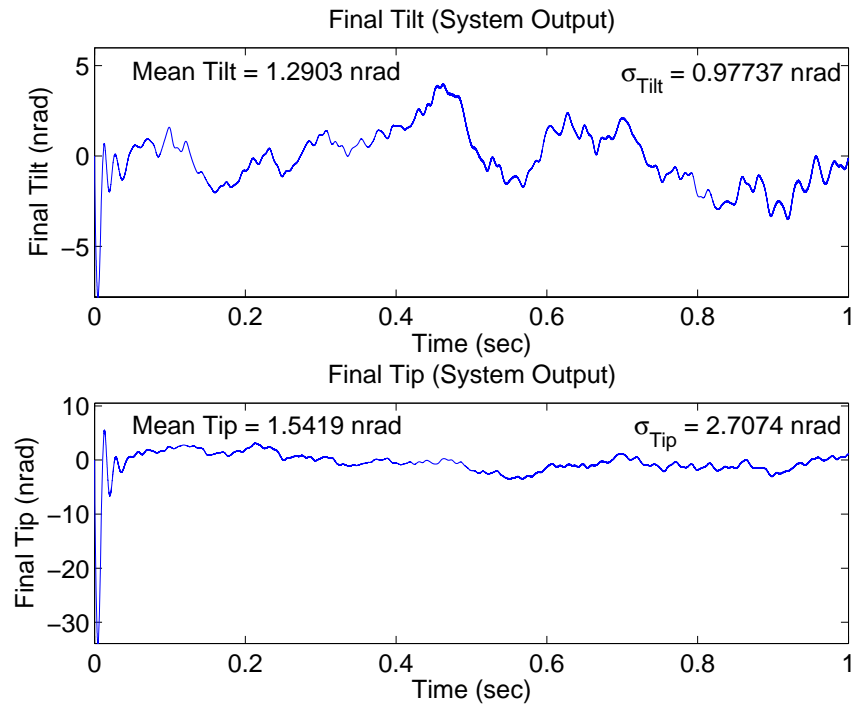


Figure 4.43 Case (7.1b): Delay 1, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

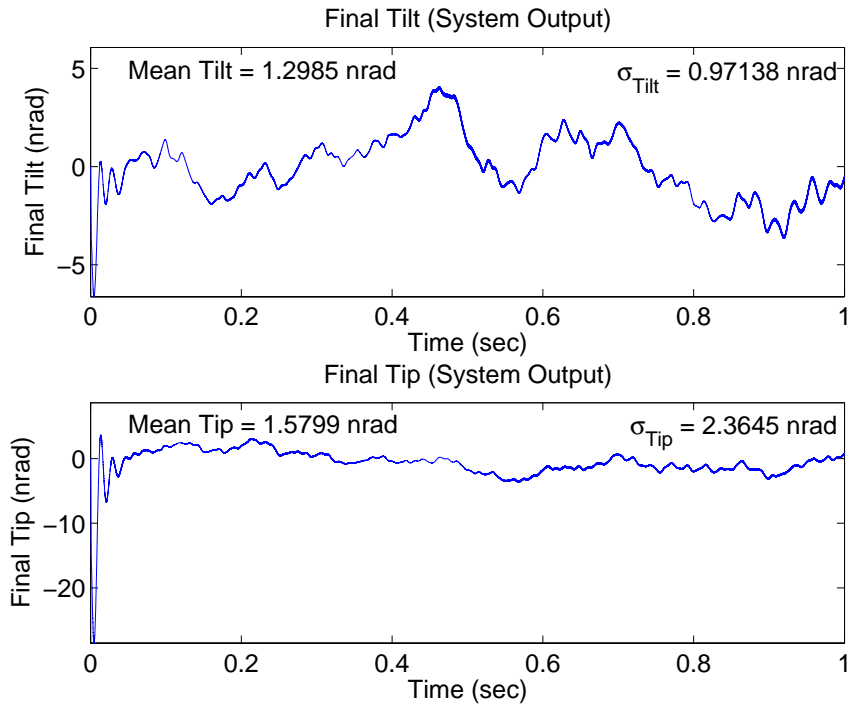


Figure 4.44 Case (7.1c): Delay 1, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

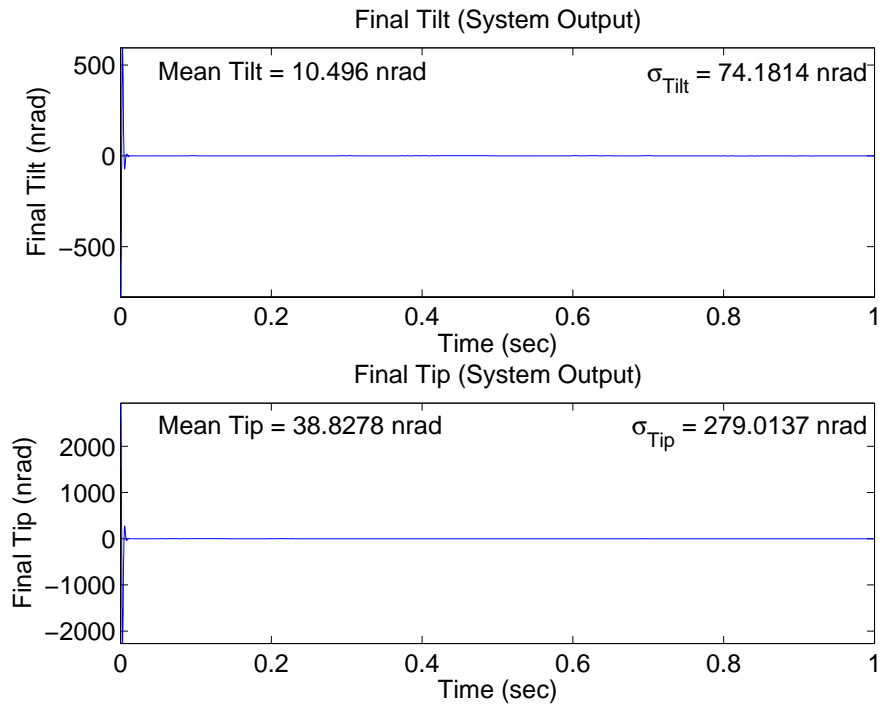


Figure 4.45 Case (8.1a): Delay 1, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

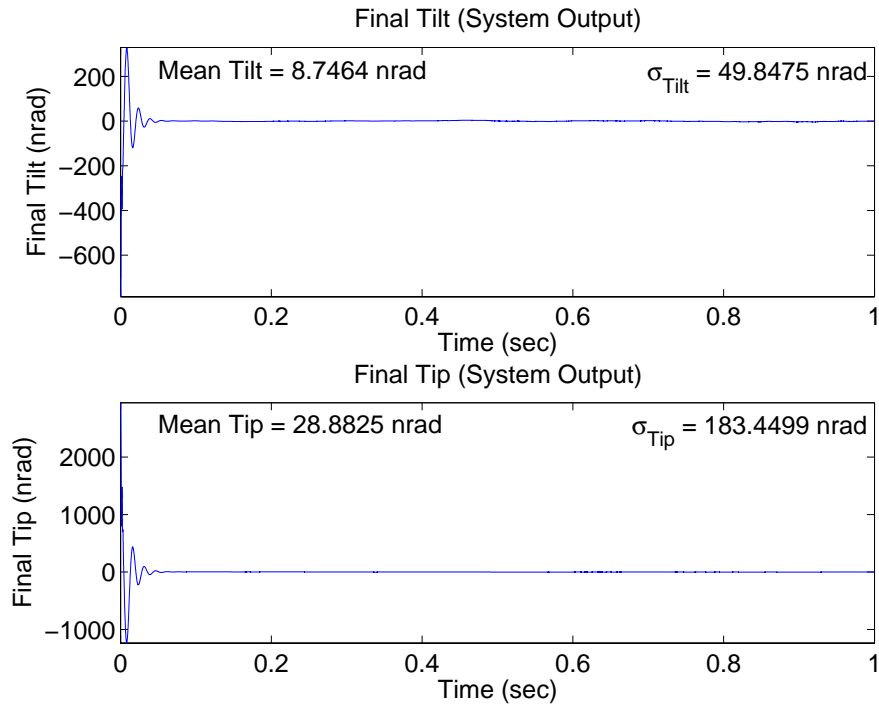


Figure 4.46 Case (8.1b): Delay 1, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

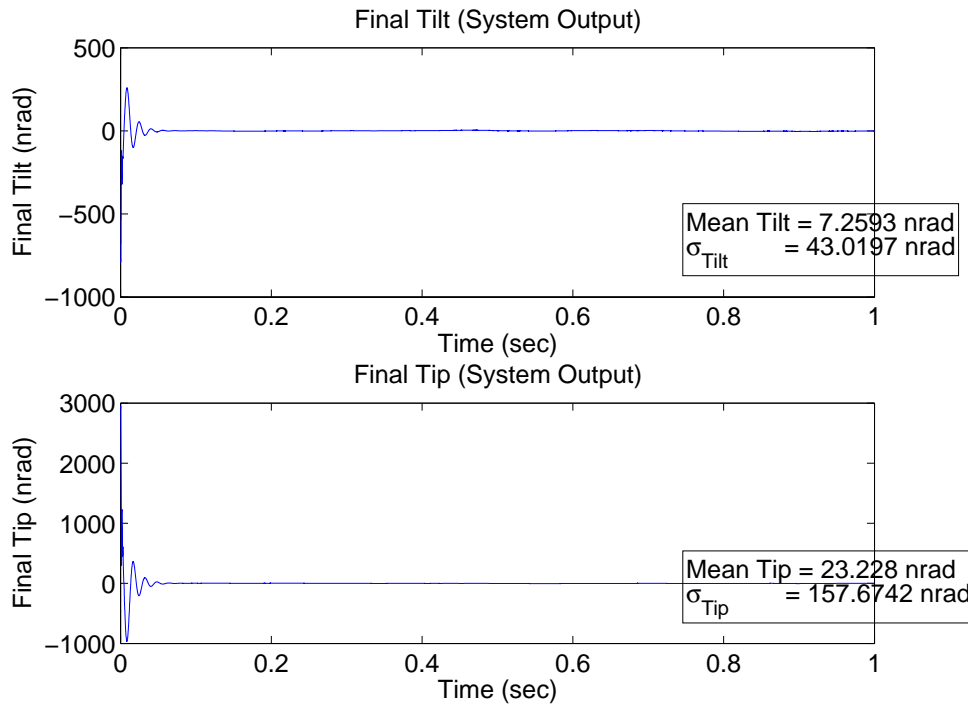


Figure 4.47 Case (8.1c): Delay 1, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

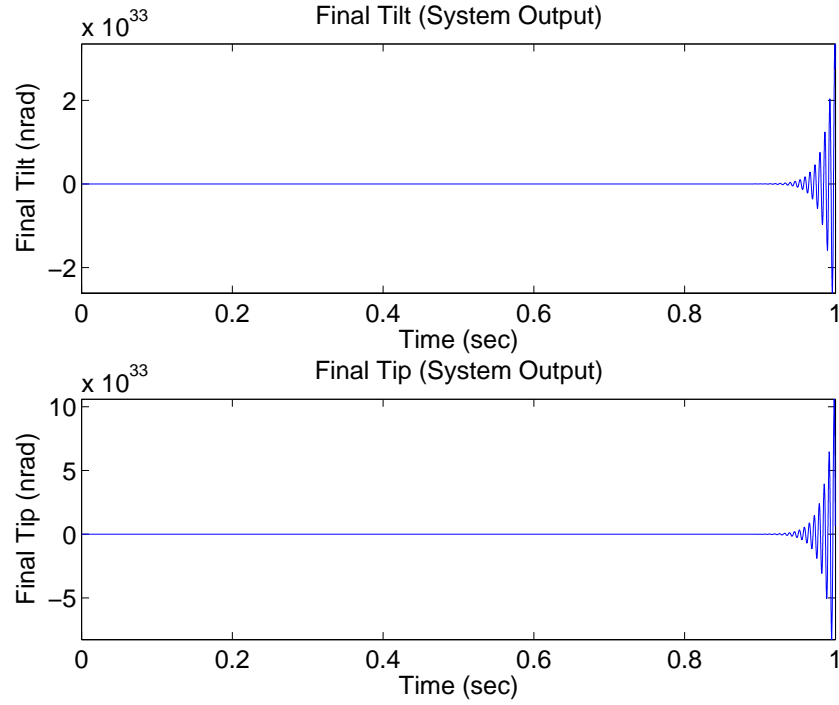


Figure 4.48 Case (7.4a): Delay 4, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

Four-Sample-Time Delay. Cases (7.4) and (8.4) simulate the control loop with a four-sample time delay. Figure 4.48 shows that Case (7.4a) resulted in an unstable performance, and Figure 4.49 shows the corresponding control inputs. The same unstable performance results from Case (8.4a). This controller design with cost function described by $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$ was able to maintain a stable system up to and including three sample time delays. Cases (7.4b), (7.4c), (8.4b), and (8.4c) resulted in stable performance, and results for a time delay of five samples are shown in the next section.

Five-Sample-Time Delay. Cases (7.5) and (8.5) simulate the control loop with a five-sample time delay. As expected, the control weighting given by subcase (a) resulted in unstable performance with a five-sample time delay. Figure 4.50 displays the performance of Case (7.5b) and Figure 4.51 displays the performance of Case (7.5c). The mean and standard deviation values in Cases (7.5b) and (7.5c) slightly increase from Cases (7.1b) and (7.1c), but the increase in each value from the nominal Cases (1b) and

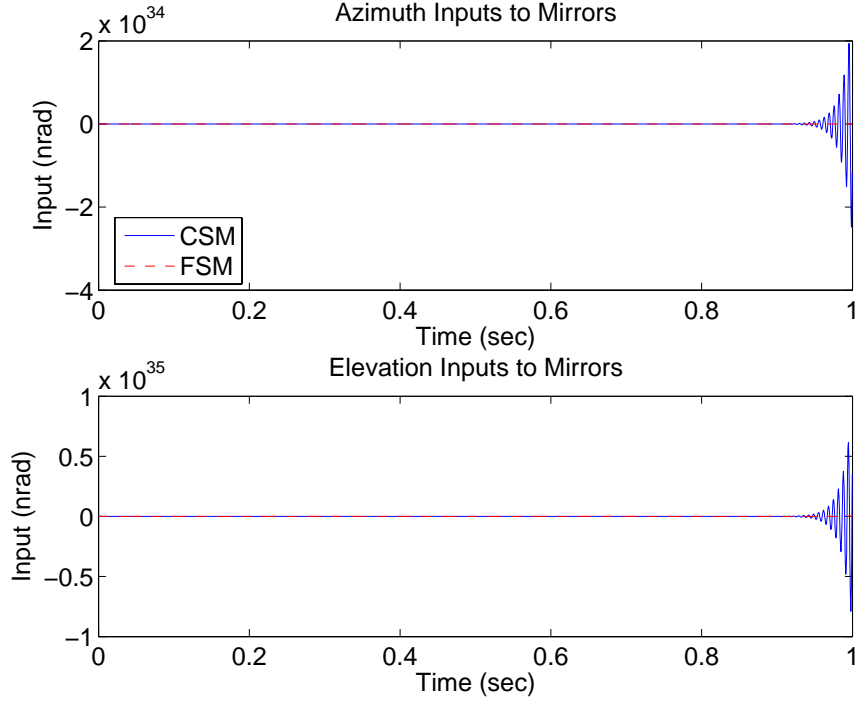


Figure 4.49 Case (7.4a): Delay 4, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

(1c) is still less than 1 nrad. The most significant change seen in Figures 4.50 and 4.51 is the increase in oscillation frequency and amplitude as well as settling time of the initial transient period. The control inputs are identical to Cases (1b) and (1c), and the results of Cases (8.5b) and (8.5c) show the same characteristics and trends as Cases (7.5b) and (7.5c) just analyzed. It is clear that the induced latency in the feedback-control is resulting in increased transients and effectively making the system appear more "sluggish," but stable.

Nine-Sample-Time Delay. Cases (7.9) and (8.9) simulate the control loop with a nine-sample time delay. As expected, the control weighting given by subcase (a) resulted in unstable performance with a nine-sample time delay. Figure 4.52 shows that Case (7.9b) also resulted in an unstable performance, and Figure 4.53 shows the corresponding control inputs. The same unstable performance results from Case (8.9b). This controller design with cost function described by subcase (b), $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$, was able to maintain a stable system up to and including eight sample time delays.

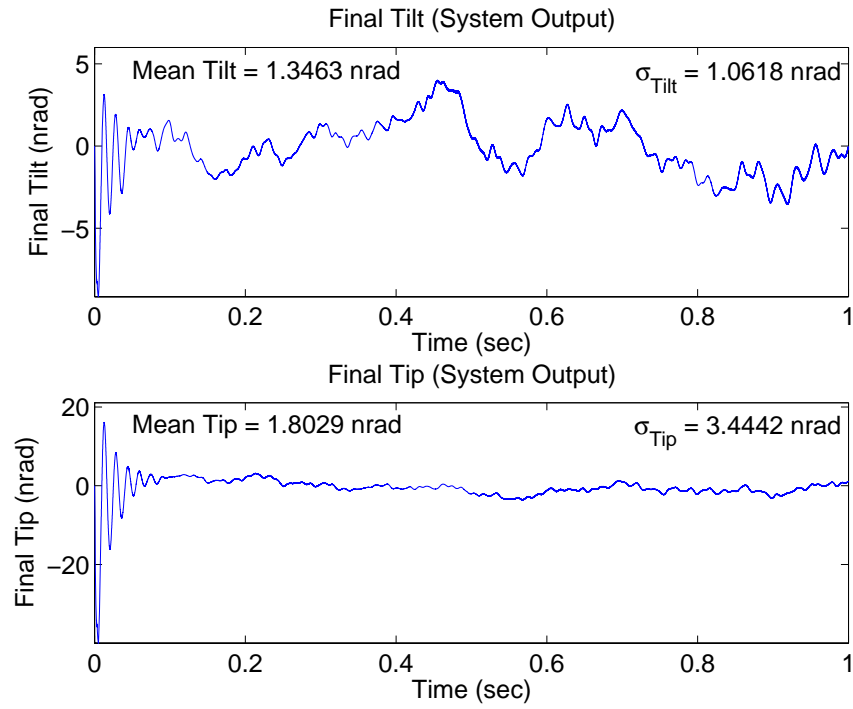


Figure 4.50 Case (7.5b): Delay 5, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

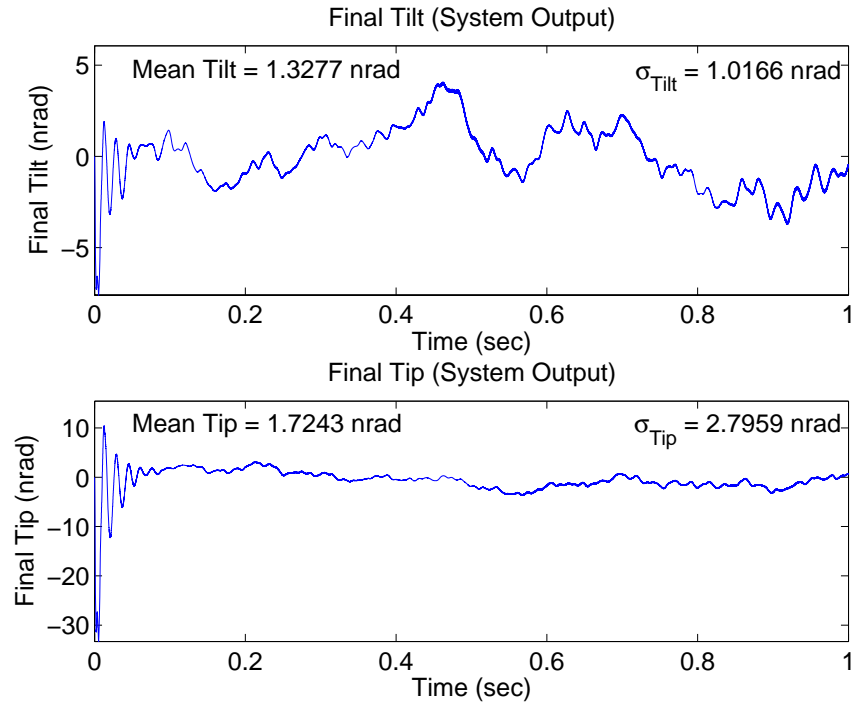


Figure 4.51 Case (7.5c): Delay 5, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

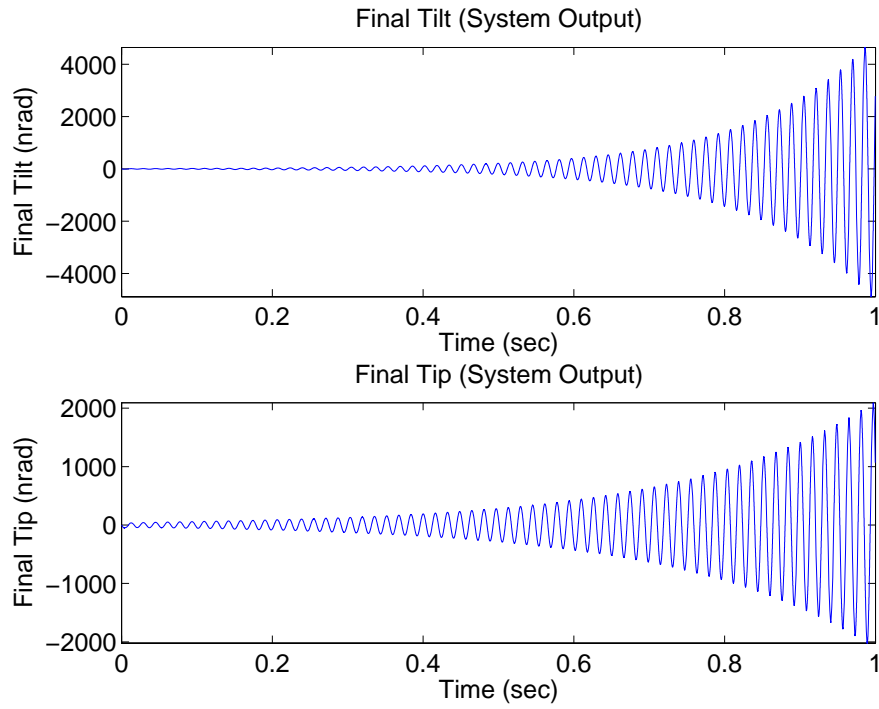


Figure 4.52 Case (7.9b): Delay 9, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

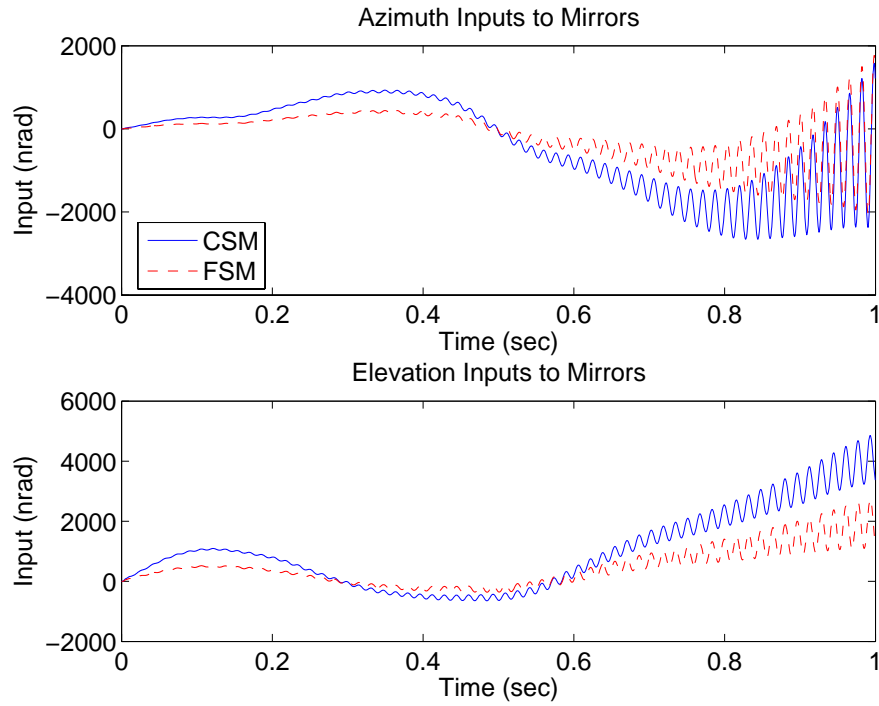


Figure 4.53 Case (7.9b): Delay 9, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

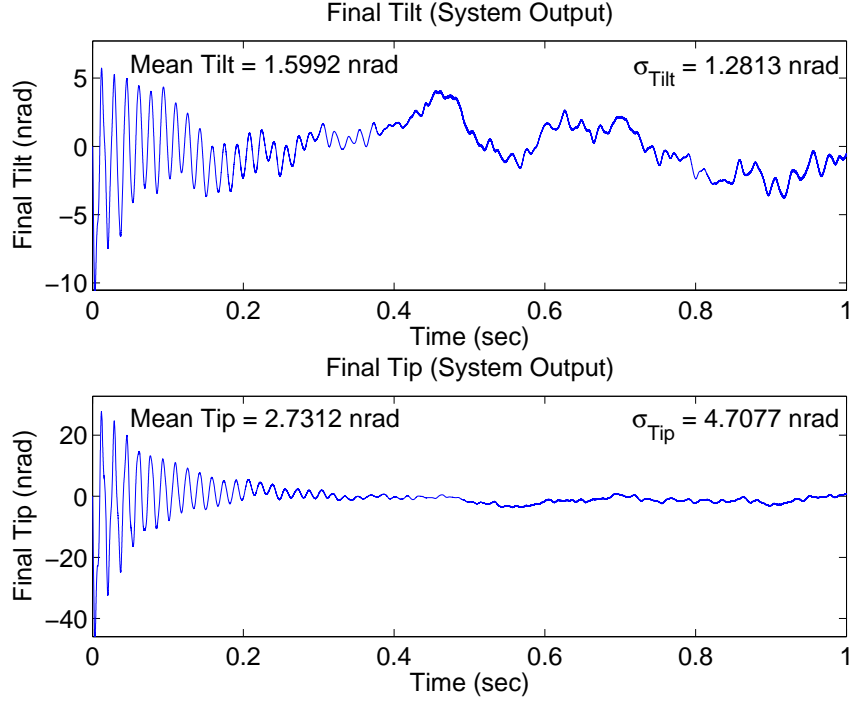


Figure 4.54 Case (7.9c): Delay 9, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

For Case (7.9c), Figure 4.54 shows the non-acquisition final tilt and tip values and Figure 4.55 shows the corresponding control inputs. The mean and standard deviation values increase by only 1 to 2 nrad as compared to Case (7.5c), and increase by only 1 to 3 nrad compared to the nominal Case (1c). The control inputs remain identical to Case (1c). For Case (8.9c), Figure 4.56 shows the acquisition final tilt and tip values and Figure 4.57 shows the corresponding control inputs. The acquisition mean and standard deviation values increase by 58 and 68 nrad respectively for the azimuth axis, and by 172 and 245 nrad respectively for the elevation axis. In both Cases (7.9c) and (8.9c), the number and amplitude of oscillations, as well as the settling time, cause the increase in mean and standard deviation. The oscillation amplitude values in the elevation axis are greater than those in the azimuth axis due to larger initial disturbance values, as shown in Figures 4.3 and 4.15, leading to the higher mean and standard deviation values in the elevation axis. The same oscillatory transient behavior can be seen in the control inputs of Case (8.9c).

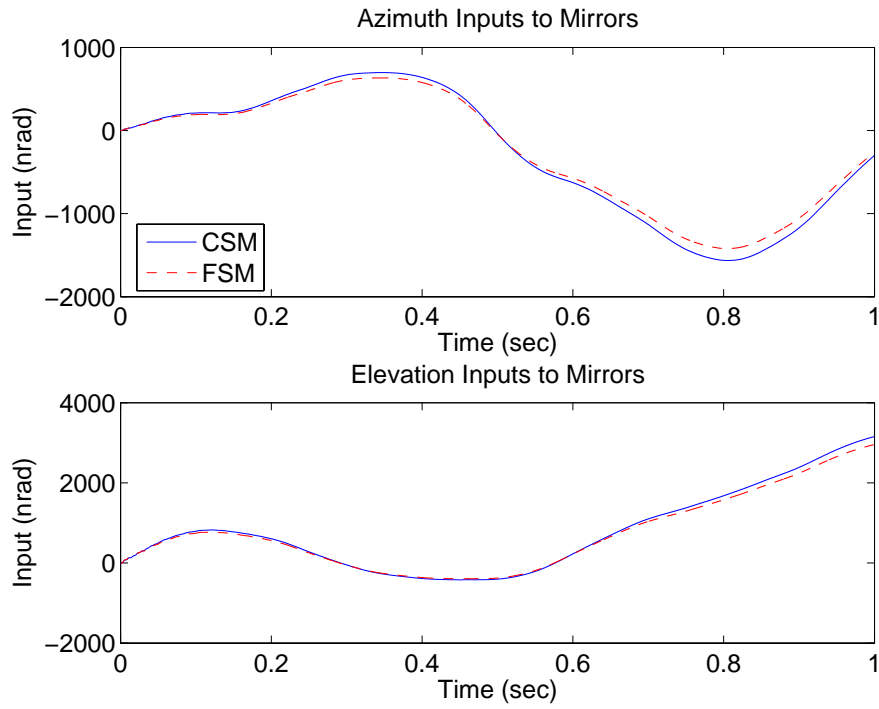


Figure 4.55 Case (7.9c): Delay 9, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

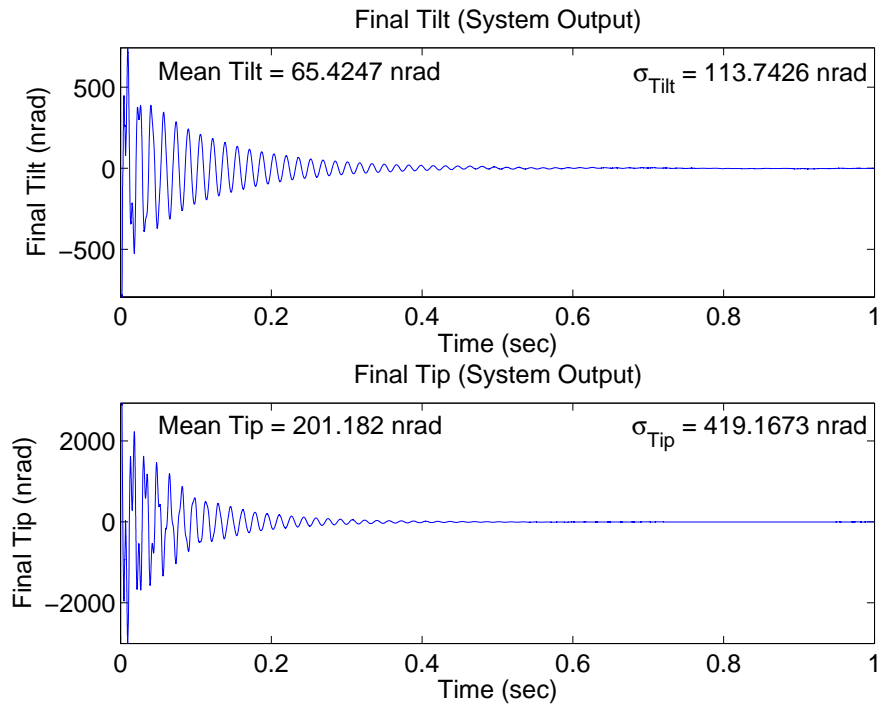


Figure 4.56 Case (8.9c): Delay 9, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

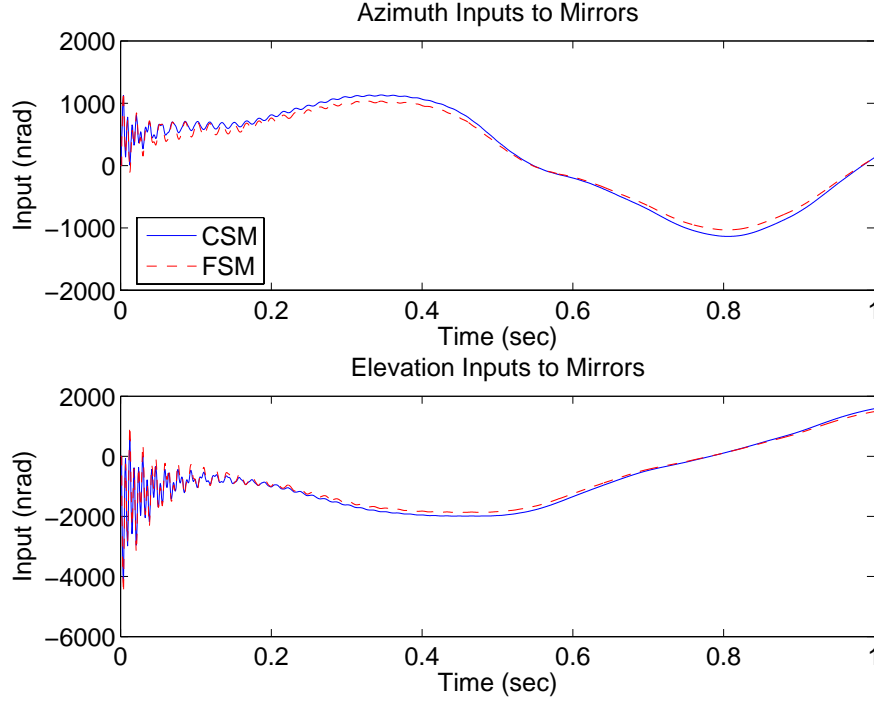


Figure 4.57 Case (8.9c): Delay 9, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

Ten-Sample-Time Delay. Cases (7.10) and (8.10) simulate the control loop with a ten-sample time delay. As expected, the control weighting given by subcases (a) and (b) resulted in unstable performance with a ten-sample time delay. Figure 4.58 shows that Case (7.10c) resulted in an unstable performance in the azimuth axis, and Figure 4.59 shows the corresponding control inputs. Performance in the elevation axis remains stable, but the oscillations last the entire simulation time and the mean and standard deviation increase as a result. The non-acquisition control inputs still appear the same as in Case (1), especially the elevation inputs, although small oscillations on the azimuth axis CSM and FSM inputs can be seen. As expected, the same type of performance occurs in Case (8.10c) as shown in Figure 4.60. Figure 4.61 displays the acquisition control inputs with same oscillatory behavior seen on the final tilt and tip. This controller design with cost function described by subcase (c), $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$, was able to maintain a stable system up to and including nine sample time delays.

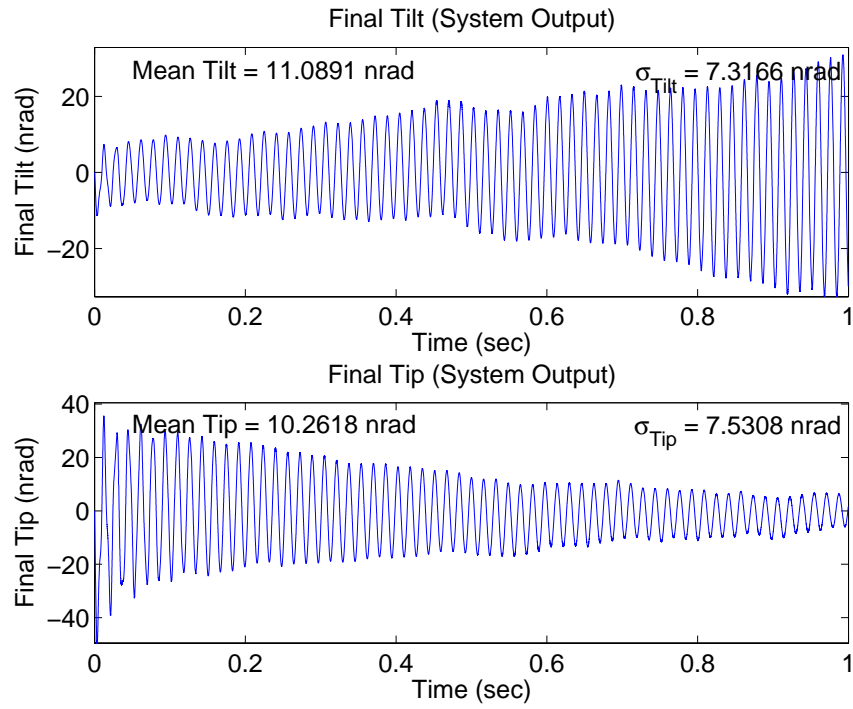


Figure 4.58 Case (7.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

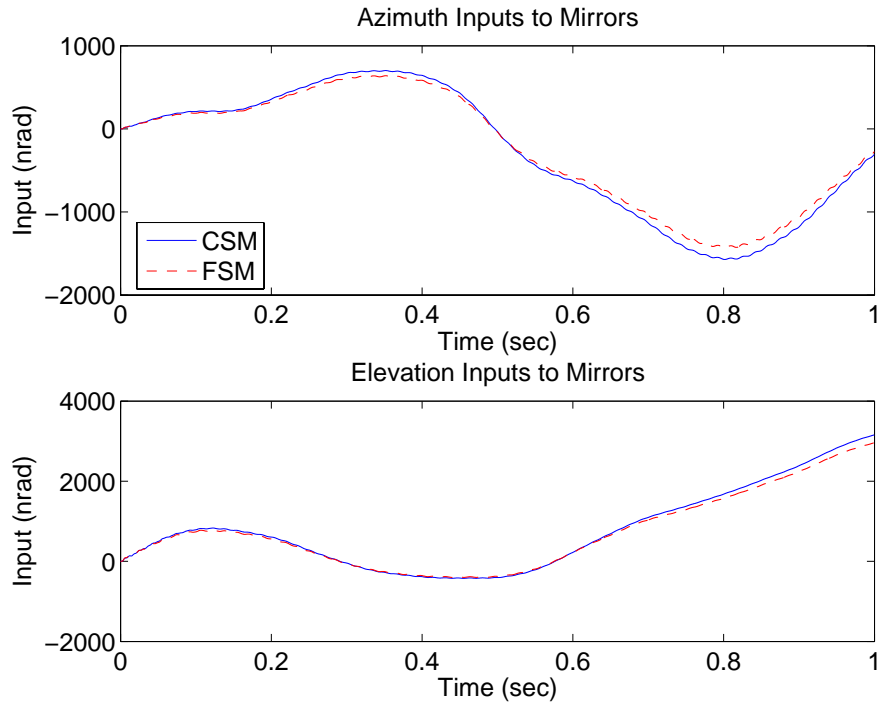


Figure 4.59 Case (7.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

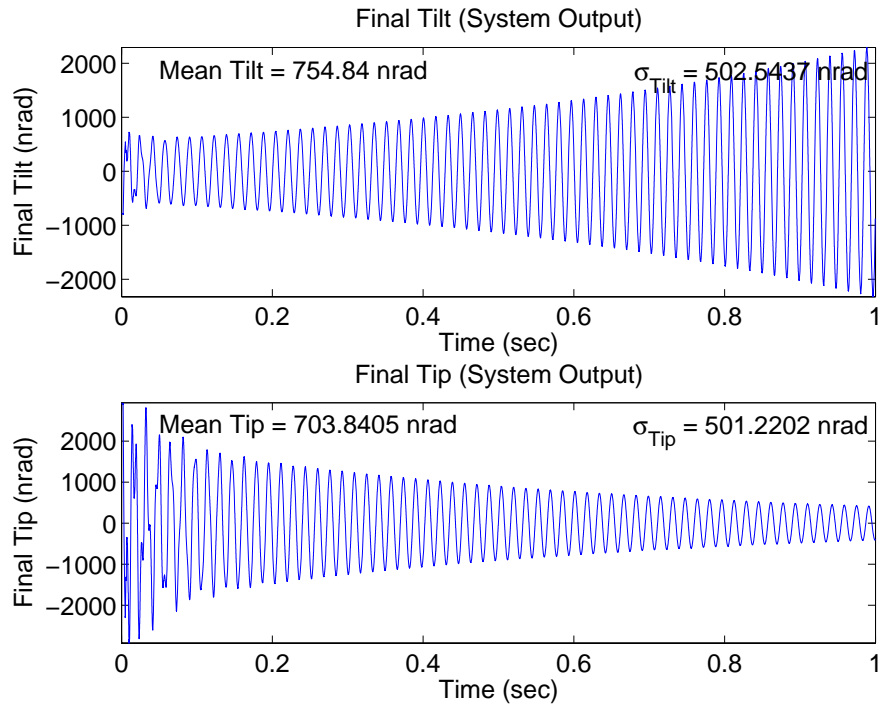


Figure 4.60 Case (8.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

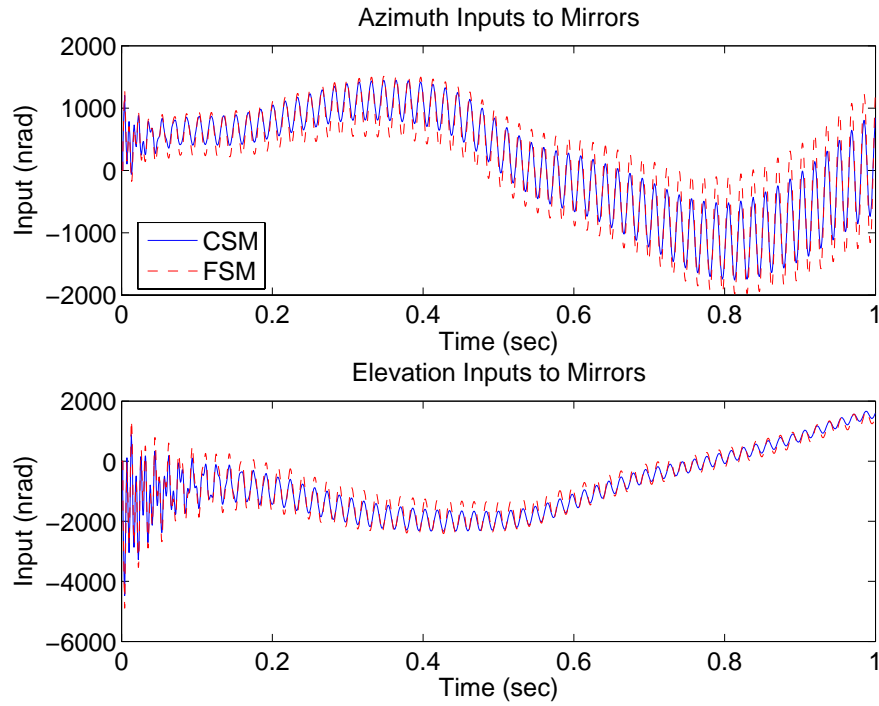


Figure 4.61 Case (8.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

4.2.1.5 *Cases (9) and (10).* The purpose of Cases (9) and (10) is to evaluate the performance and robustness of the control system to various system delay times with measurement saturation. The delay times and saturation values are identical to the values used in Cases (5) through (8), so the goal of the analysis is to determine if the robustness properties seen in Cases (7) and (8) change with the addition of measurement saturation. Recall that no change in performance was observed in Case (5), and overall azimuth axis performance in Case (6) slightly improved while the tip mean and standard deviation values in Case (6) increased slightly. As in Case (5) compared to Case (1), in which the saturation had no effect, the results from Case (9) are identical to Case (7) for all delay times and thus are not shown. Therefore, the following sections present the results of Case (10).

One-Sample-Time Delay with Measurement Saturation. Case (10.1) simulates the control loop with a one-sample time delay and measurement saturation from the pyramid sensor. Figures 4.62 to 4.64 show the acquisition final tilt and tip values and Figures 4.65 to 4.67 show the corresponding control inputs. The results of Case (10.1) as compared to Case (8.1) show the same trends seen in Case (6) compared to the nominal Case (2). Compared to Case (8.1), the mean tilt and standard deviation values improve minimally in Case (10.1a) and increase in Cases (10.1b) and (10.1c). The mean tip values increase by 16 to 18 nrad and standard deviation values increase by 40 (Case (10.1a)) to 86 nrad (Case (10.1c)). Case (10.1c) still shows the best performance and Case (10.1a) the worst performance. The transient characteristics induced from the time delay and saturation are evident in Case (10) through the reduced amplitudes and extended settling times in the final tilt/tip and control input values.

Three-Sample-Time Delay through Ten-Sample-Time Delay with Measurement Saturation. These time delays are grouped together in one section due to a common trend observed in the results. The performance in Cases (10.3) through (10.10) actually improve compared to Cases (8.3) through (8.10) and the control system based on each of the three controller designs now shows marginally stable performance at the delay

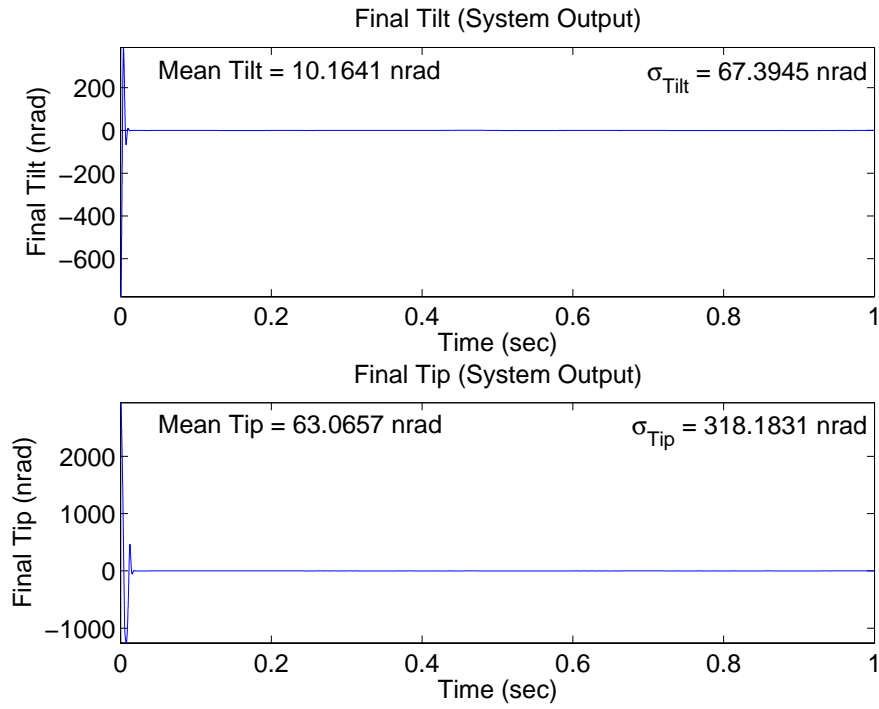


Figure 4.62 Case (10.1a): Delay 1, $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

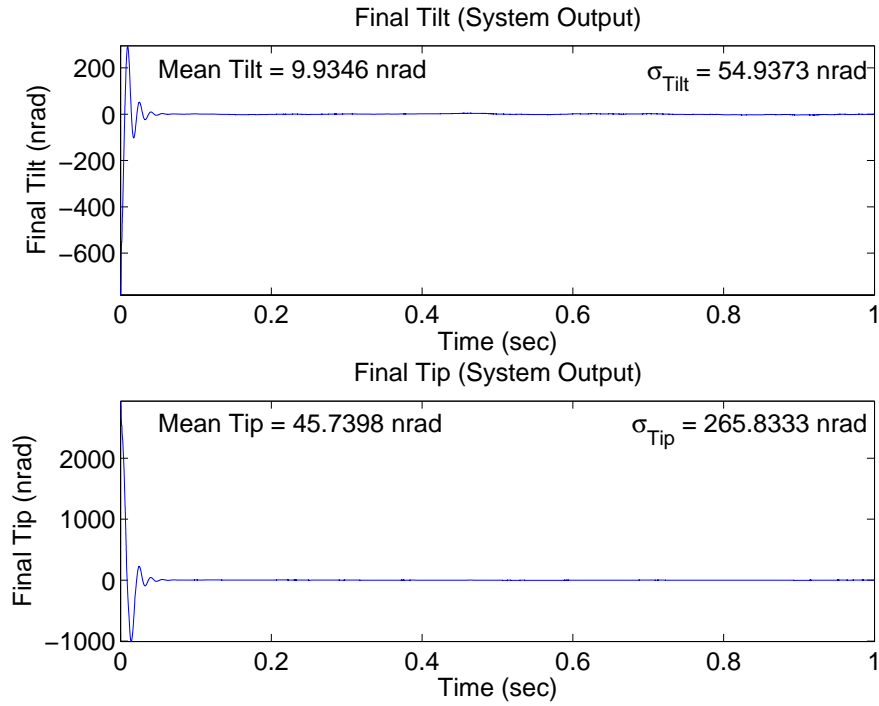


Figure 4.63 Case (10.1b): Delay 1, $\mathbf{U}_{diag} = [3E5 \quad 3E5 \quad 1E4 \quad 1E4]$

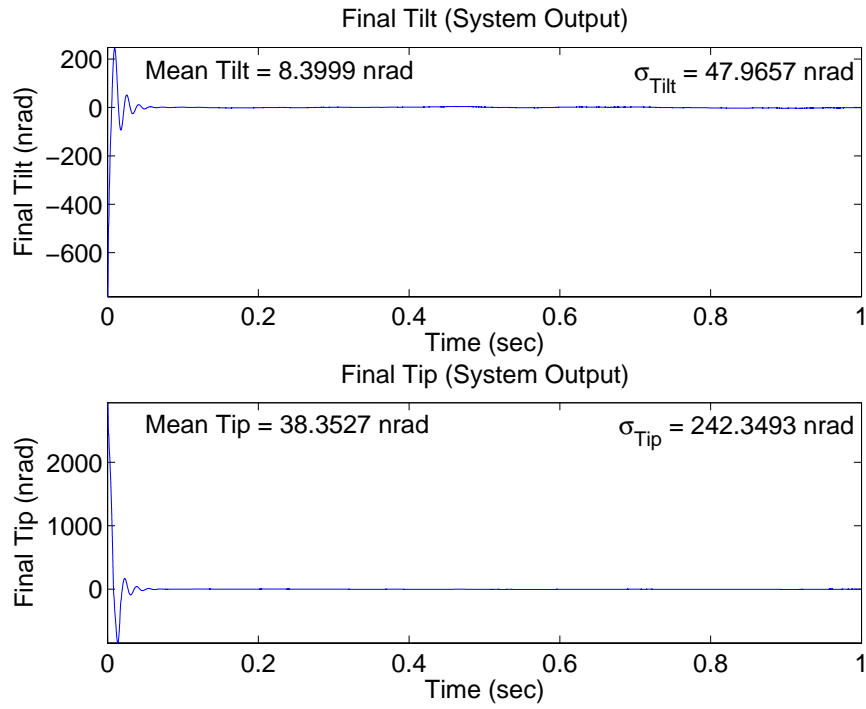


Figure 4.64 Case (10.1c): Delay 1, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

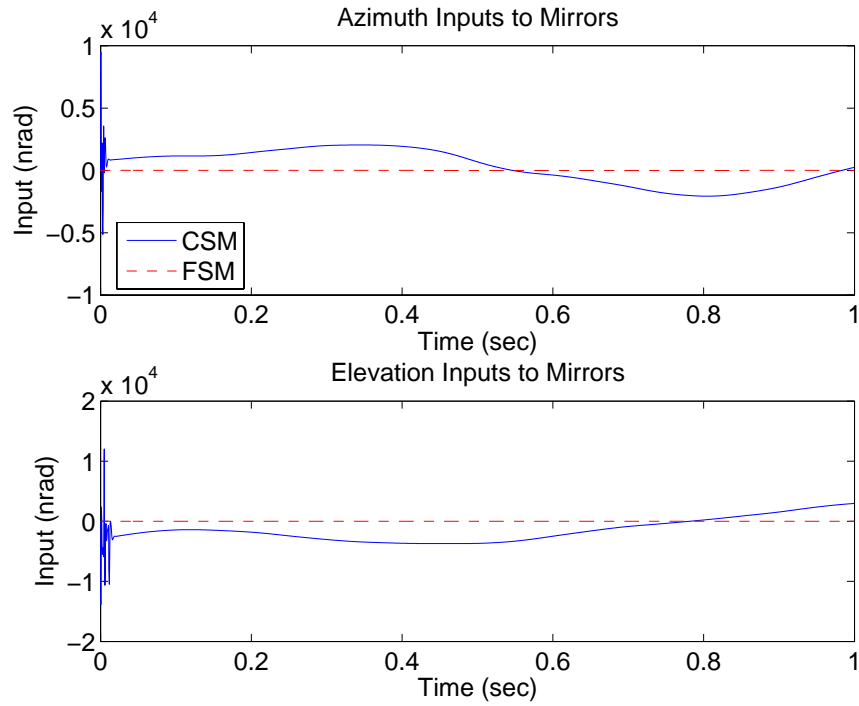


Figure 4.65 Case (10.1a): Delay 1, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

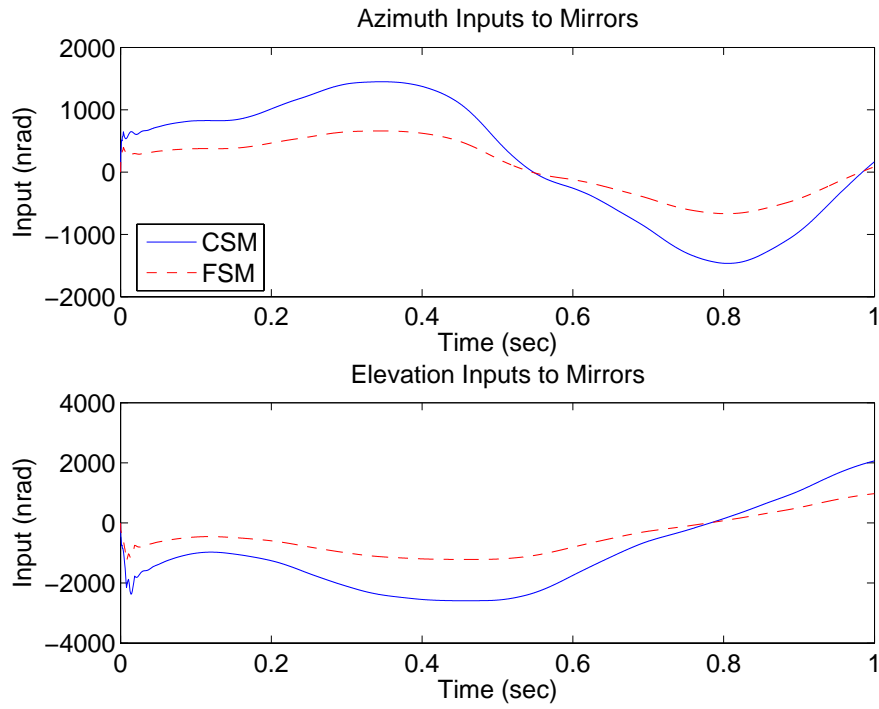


Figure 4.66 Case (10.1b): Delay 1, $\mathbf{U}_{diag} = [3E5 \ 3E5 \ 1E4 \ 1E4]$

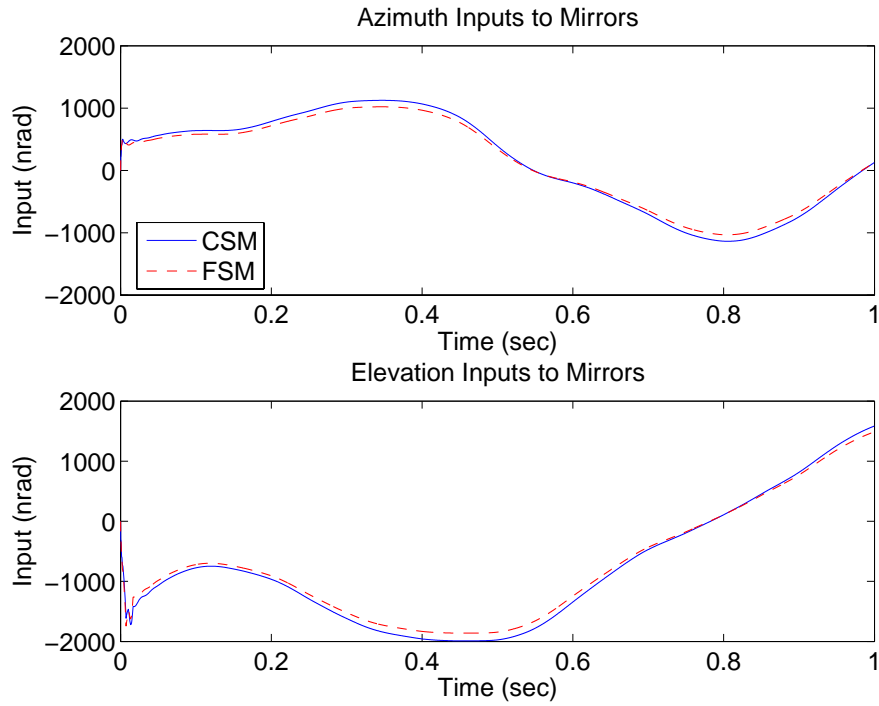


Figure 4.67 Case (10.1c): Delay 1, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

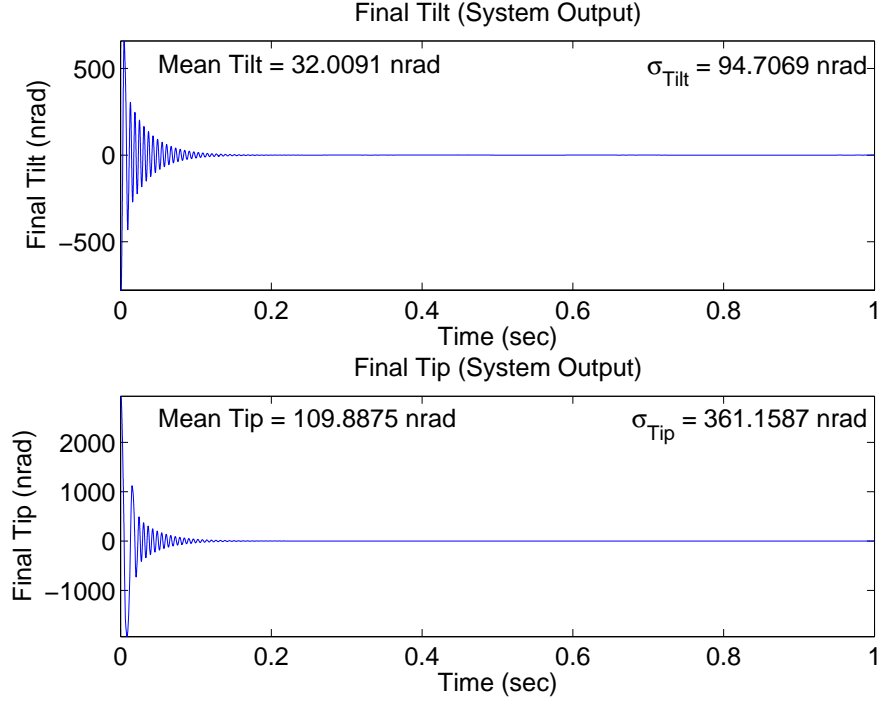


Figure 4.68 Case (10.3a): Delay 3, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

values noted in Case (8) that caused unstable performance. Selected data from Case (10) are presented in the following figures to show the improved performance.

Figure 4.68 shows the final tilt and tip for Case (10.3a) and Figure 4.69 shows the final tilt and tip for Case (8.3a) since it was not shown previously. Both axes show improvement in mean and standard deviation, especially the elevation axis. The improvement results from decreased transient amplitude and oscillation due to the pyramid sensor saturation. The settling time of the transient period remains the same in both cases. The control inputs exhibit the same characteristics and are not shown. Figures 4.70 to 4.72 show comparisons of the results from the remaining two controller designs for particular delay times to show the same performance trends just discussed. Figure 4.72 is compared to Figure 4.56. Again, the performance from Case (10) improves due to the reduced oscillation amplitude and number of oscillations, especially in the elevation axis. Figures 4.73 to 4.75 display the marginally stable performance resulting from Cases (10.4a), (10.9b), and

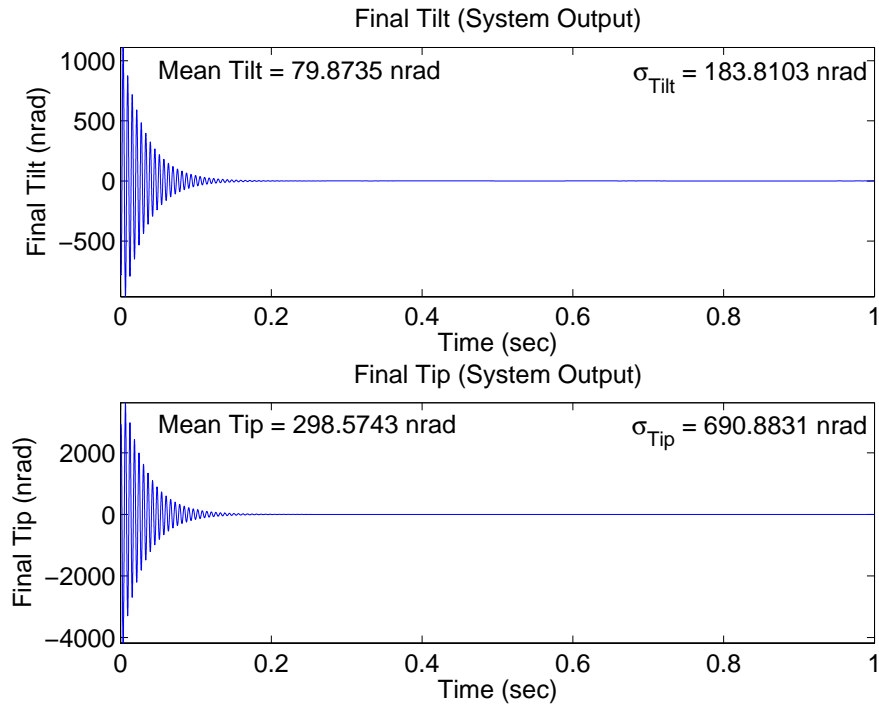


Figure 4.69 Case (8.3a): Delay 3, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

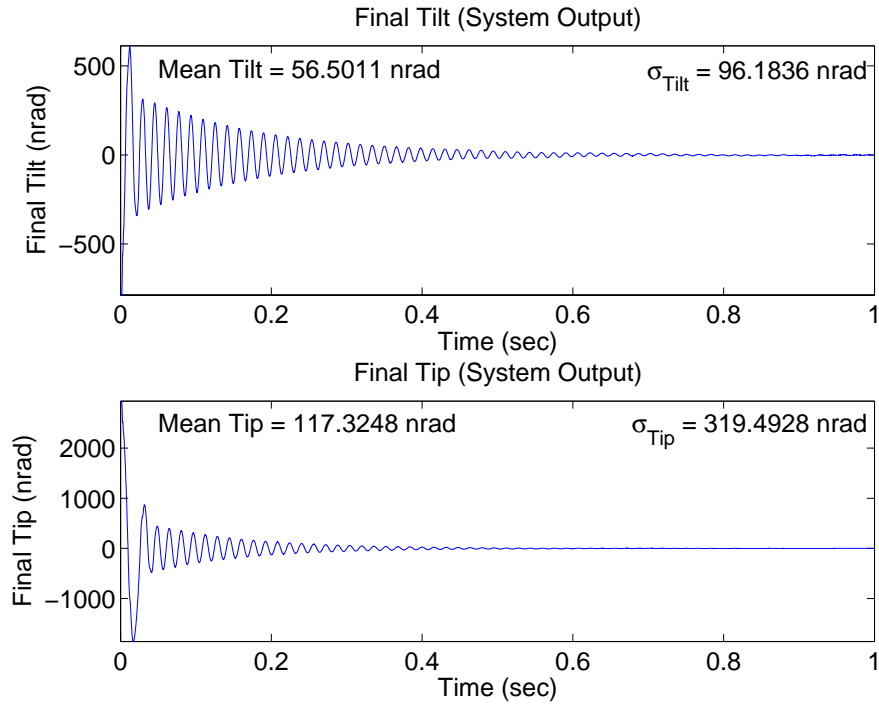


Figure 4.70 Case (10.8b): Delay 8, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

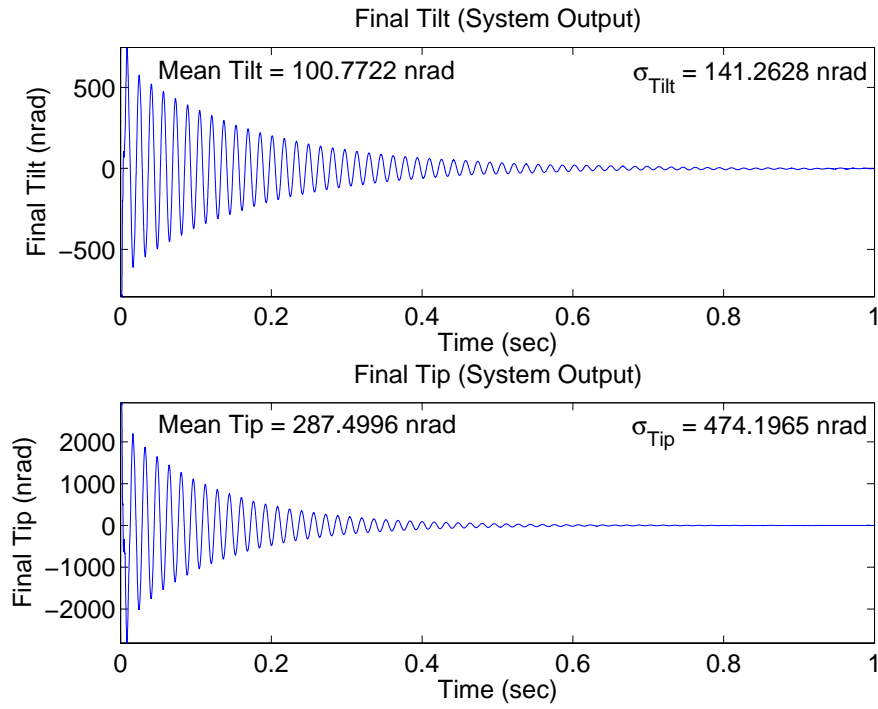


Figure 4.71 Case (8.8b): Delay 8, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

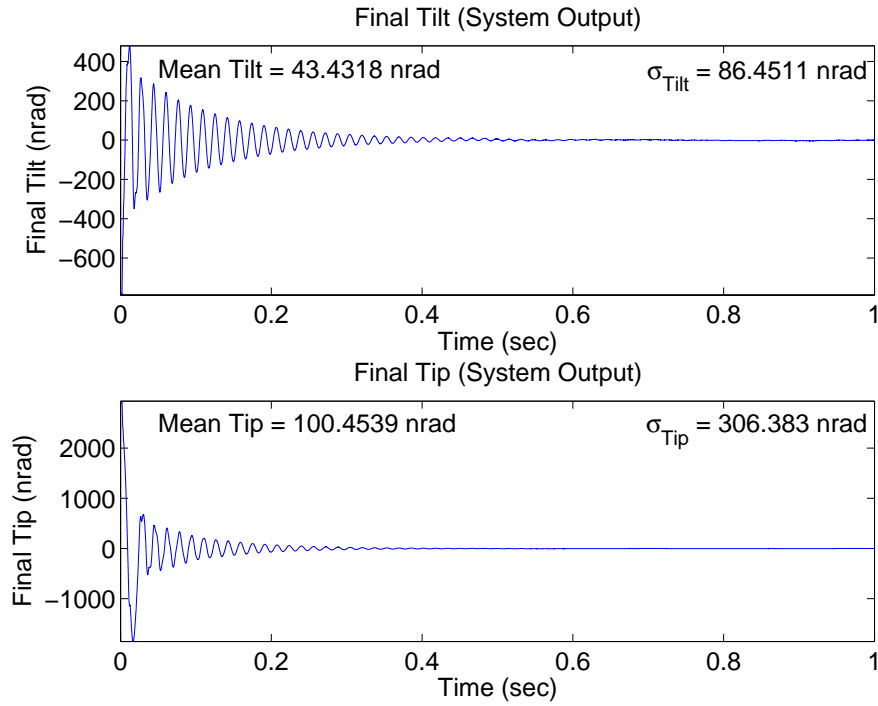


Figure 4.72 Case (10.9c): Delay 9, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

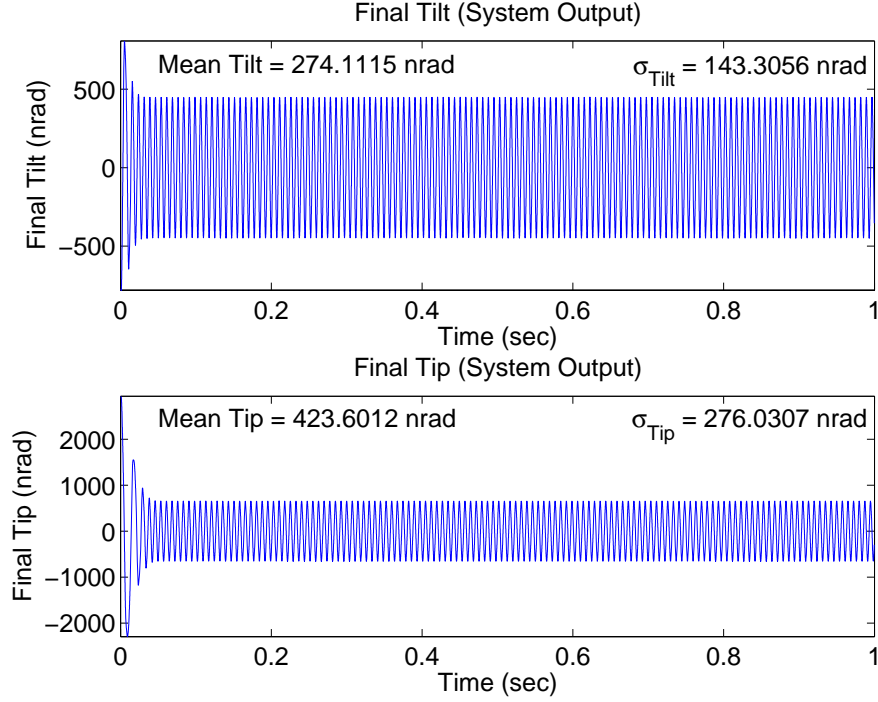


Figure 4.73 Case (10.4a): Delay 4, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

(10.10c) corresponding to the same delay times that resulted in unstable performance in Cases (8.4a), (8.9b), and (8.10c).

4.2.1.6 Cases (11) and (12). The purpose of Cases (11) and (12) is to evaluate the performance and robustness of the control system based on the previous controller designs with the LTR technique described by Equation (3.63). The q_{LTR} value in Equation (3.63) is tuned to find the value for each of the three cost functions that provides the best observed robustness to the time delays and saturation. In other words, the q_{LTR} values that kept the control system stable, while subjected to the most time delays, were chosen. The LQG/LTR control designs are then simulated in Cases (13) and (14) with no time delays and no saturation to evaluate the amount of performance potentially lost in order to increase robustness. For each of the previous three LQG cost functions, the control weighting matrices were kept the same while the state weighting matrices were replaced by the matrices based on the chosen q_{LTR} values. The chosen q_{LTR}

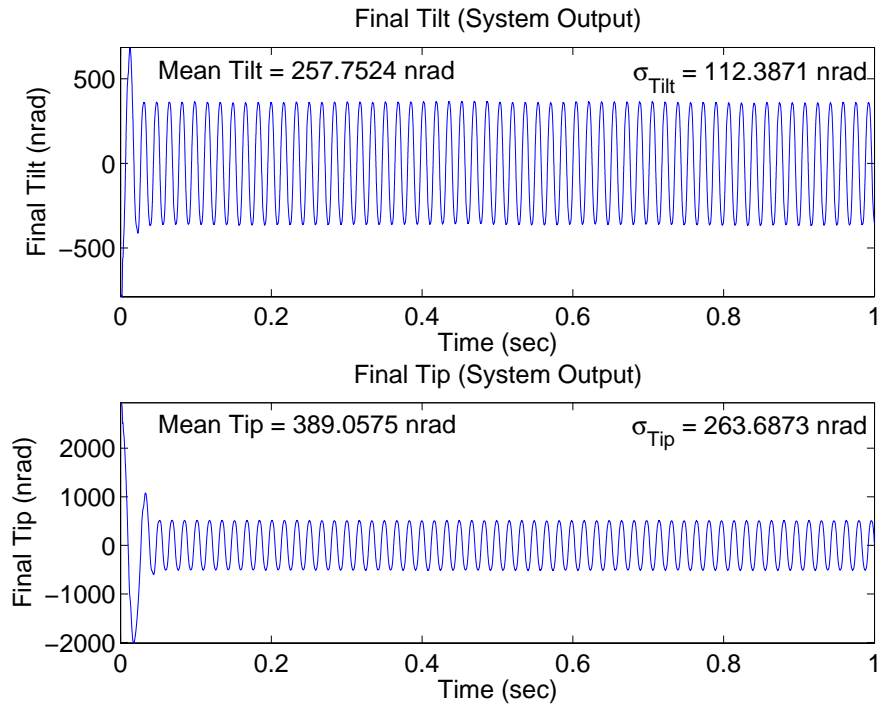


Figure 4.74 Case (10.9b): Delay 9, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

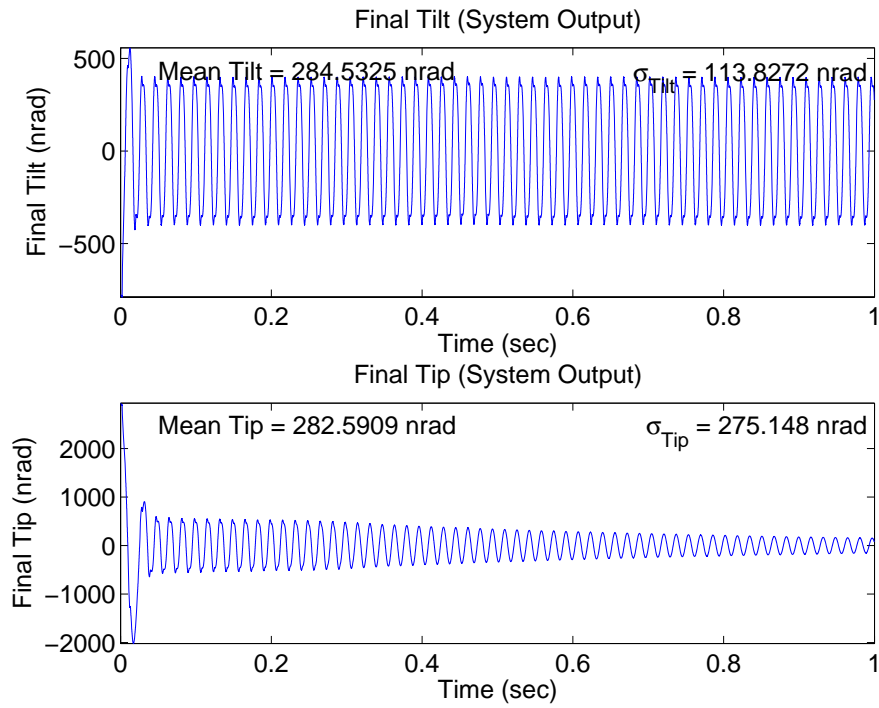


Figure 4.75 Case (10.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

values along with the corresponding control weighting matrices are

(a)	$q_{LTR} = 1E3$	$\mathbf{U}_{diag} = \begin{bmatrix} 1E0 & 1E0 & 1E4 & 1E4 \end{bmatrix}$
(b)	$q_{LTR} = 1E3$	$\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$
(c)	$q_{LTR} = 500$	$\mathbf{U}_{diag} = \begin{bmatrix} 1E4 & 1E4 & 1E4 & 1E4 \end{bmatrix}$

The new \mathbf{X} matrices formed by the LTR technique are shown in Appendix A.4.

The use of the LTR technique enables the control system based on each controller design to remain stable up to one more sample time delay than in Cases (7), (8), (9), and (10). Figures 4.76 and 4.77 show the final tilt and tip for Cases (11.4a) and (12.4a), consisting of a four-sample time delay and saturation. The control system now remains stable, unlike Cases (7.4a), (8.4a), (9.4a), and (10.4a). Compared to Cases (9.3a) and (10.3a), consisting of a three-sample time delay and saturation, Case (11.4a) results in nearly identical mean and standard deviation values and the values in Case (12.4a) improve, especially in the elevation axis.

Figures 4.78 and 4.79 show the final tilt and tip for Cases (11.9b) and (12.9b), consisting of a nine-sample time delay and saturation. The control system now remains stable, unlike Cases (7.9b), (8.9b), (9.9b), and (10.9b). Compared to Cases (9.8b) and (10.8b), consisting of an eight-sample time delay and saturation, the mean and standard deviation values of Cases (11.9b) and (12.9b) are less, especially in Case (12.9b).

Figures 4.80 and 4.81 show the final tilt and tip for Cases (11.10c) and (12.10c), consisting of a ten-sample time delay and saturation. The control system now remains stable, unlike Cases (7.10c), (8.10c), (9.10c), and (10.10c). The mean and standard deviation values in Cases (11.10c) and (12.10c), especially (12.10c), are greater than the values in Cases (9.9c) and (10.9c), which consist of a nine-sample time delay and saturation.

4.2.1.7 Cases (13) and (14). The purpose of Cases (13) and (14) is to evaluate the performance of the LQG/LTR designs under nominal conditions of no delay and no saturation, and compare the results to Cases (1) and (2). Figures 4.82 to 4.87 show the final tilt and tip from Cases (13) and (14). Comparing the results to Cases (1)

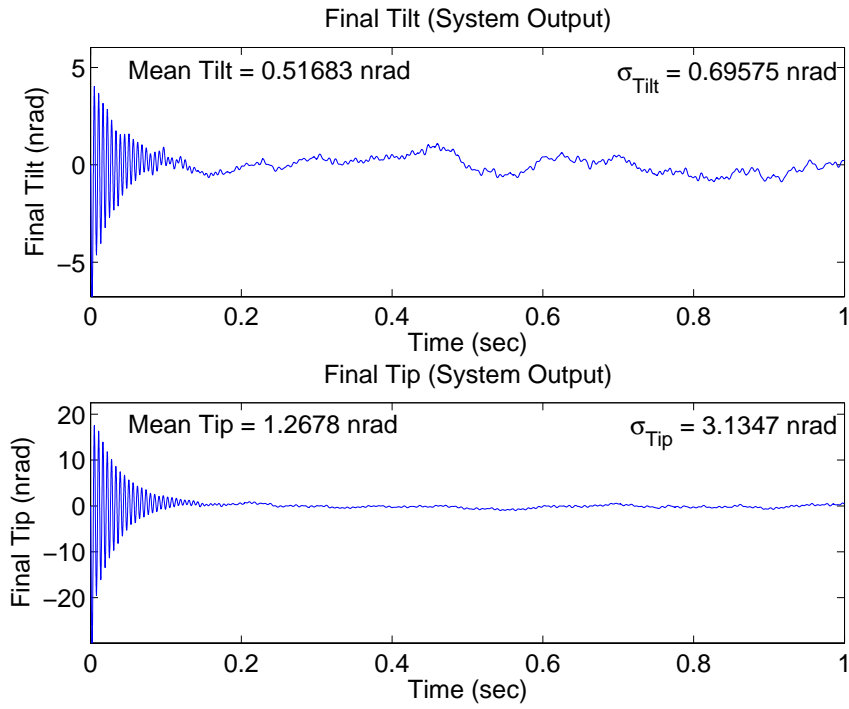


Figure 4.76 Case (11.4a): Delay 4, $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

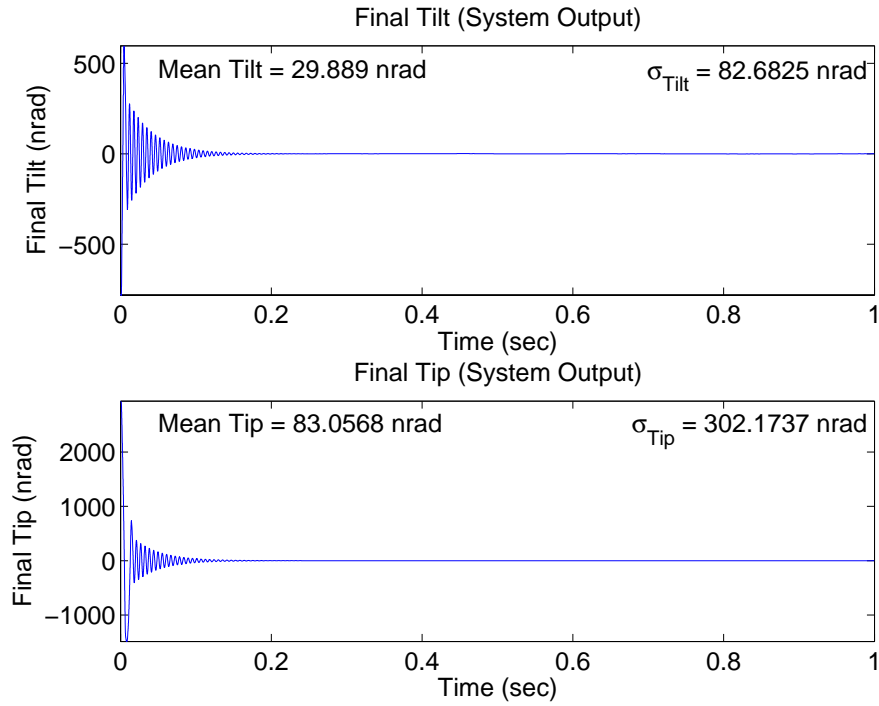


Figure 4.77 Case (12.4a): Delay 4, $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

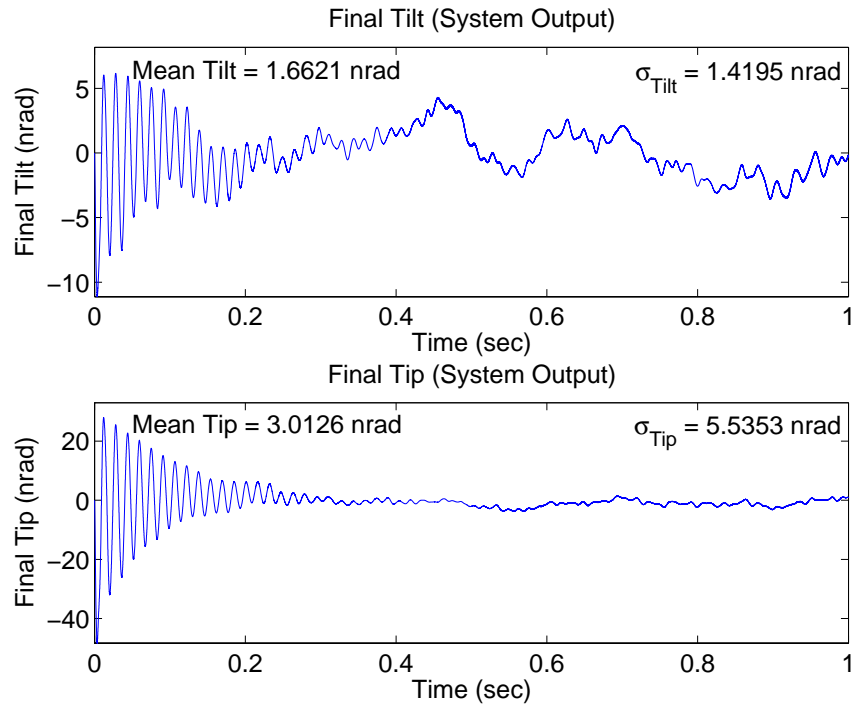


Figure 4.78 Case (11.9b): Delay 9, $\mathbf{U}_{diag} = [3E5 \ 3E5 \ 1E4 \ 1E4]$

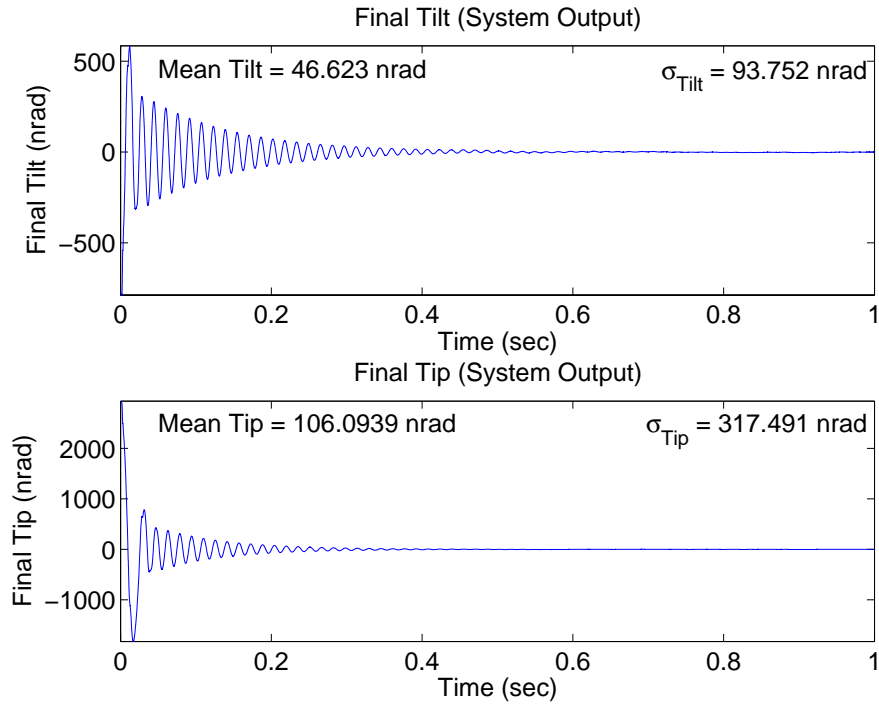


Figure 4.79 Case (12.9b): Delay 9, $\mathbf{U}_{diag} = [3E5 \ 3E5 \ 1E4 \ 1E4]$

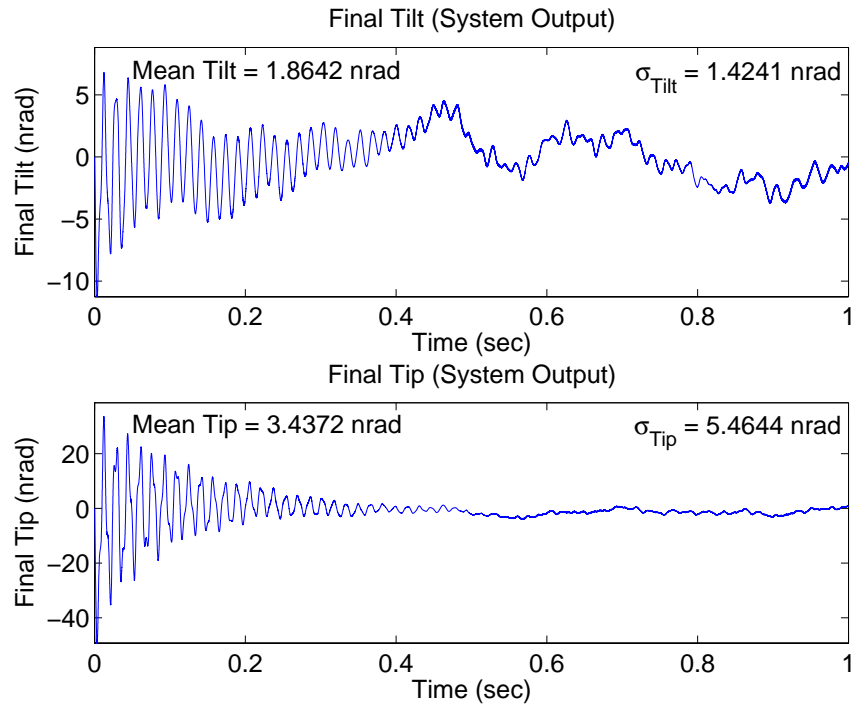


Figure 4.80 Case (11.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

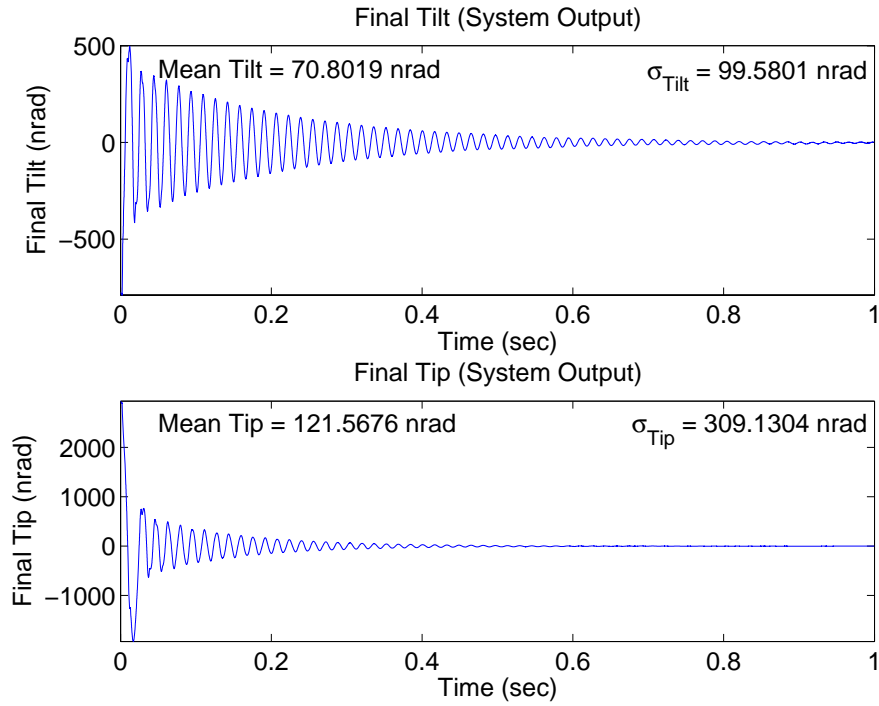


Figure 4.81 Case (12.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

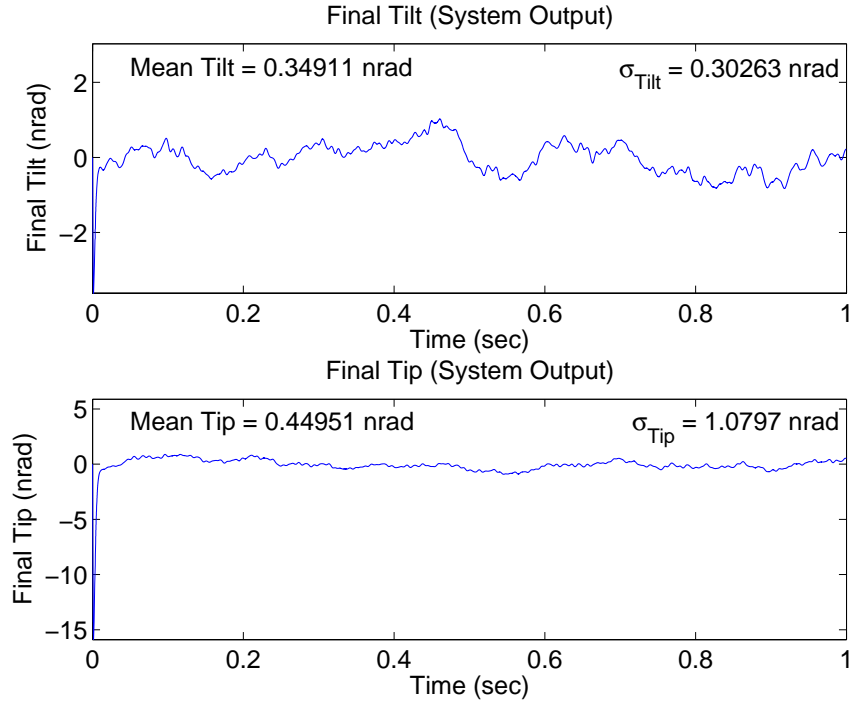


Figure 4.82 Case (13a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

and (2) demonstrates that there is no significant degradation in performance when using the LQG/LTR designs. The mean tilt and tip values in Case (13) increase by no more than 0.1 nrad and the standard deviation slightly decreases. The performance in Case (14) actually improves over Case (2), with mean values that decrease by as much as 3.1 nrad and standard deviation values that decrease by as much as 24 nrad. The control input values are not shown since they do not differ from Cases (1) and (2).

4.2.1.8 Steering Mirror Outputs. Figures 4.88 to 4.90 show the outputs of the CSM and FSM for Case (1). Note that the FSM tilt scaling in Figure 4.88 is much smaller than in the other two figures since most of the control authority in Case (1a) is given to the CSM. The control authority given by each design is evident from the decrease in CSM tilt and increase in FSM tilt from subcase (a) to subcase (c). The more drastic changes in FSM tilt from sample time to sample time shown in Figure 4.88 are also present in Figures 4.89 and 4.90, but cannot be seen due to the larger scale. The elevation outputs have the same characteristics and the plots are representative of all baseline cases.

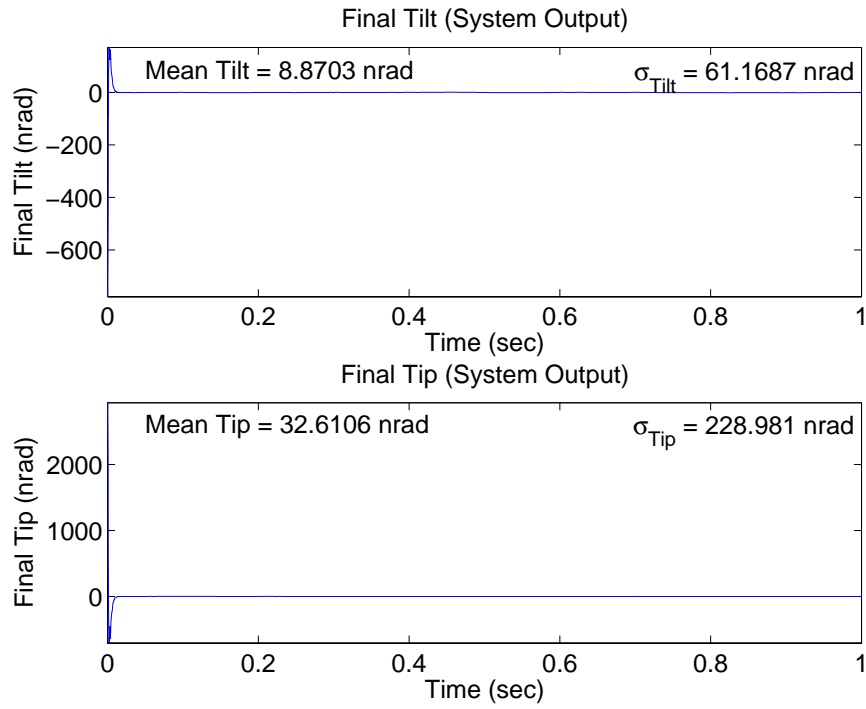


Figure 4.83 Case (14.a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

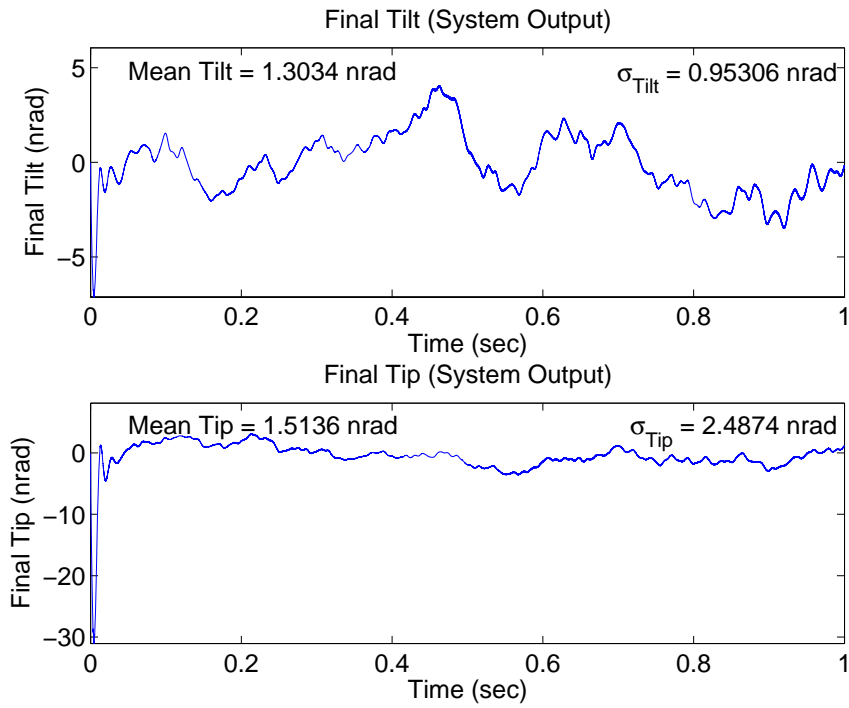


Figure 4.84 Case (13b): $\mathbf{U}_{diag} = [3E5 \ 3E5 \ 1E4 \ 1E4]$

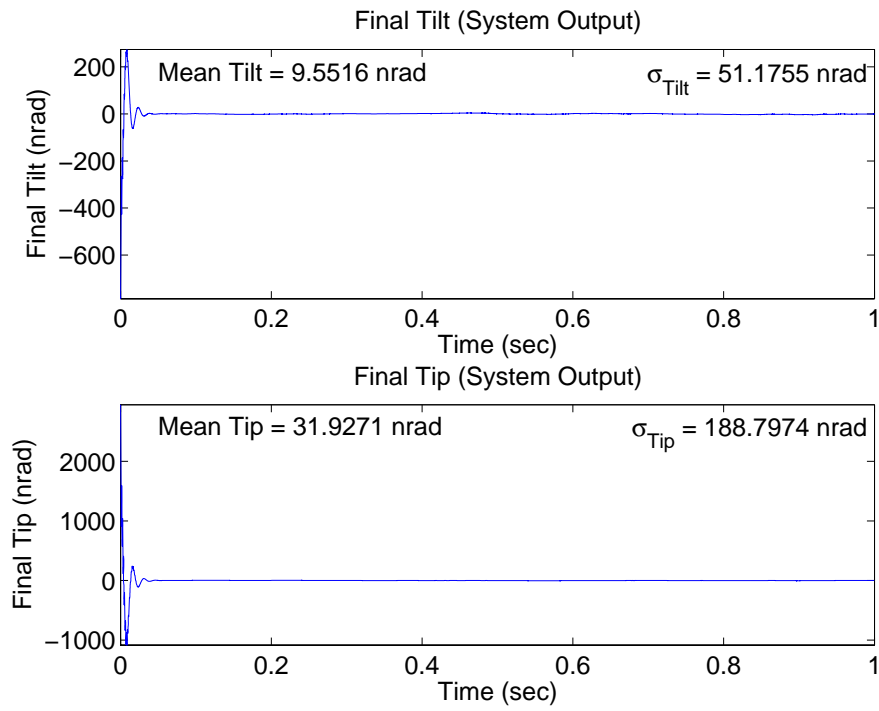


Figure 4.85 Case (14b): $\mathbf{U}_{diag} = [3E5 \ 3E5 \ 1E4 \ 1E4]$

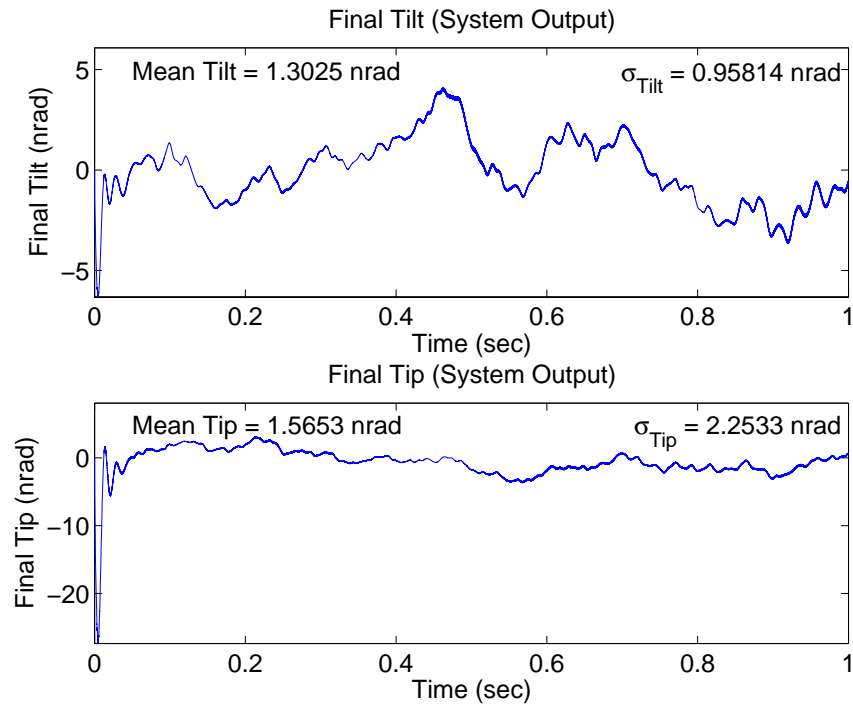


Figure 4.86 Case (13c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

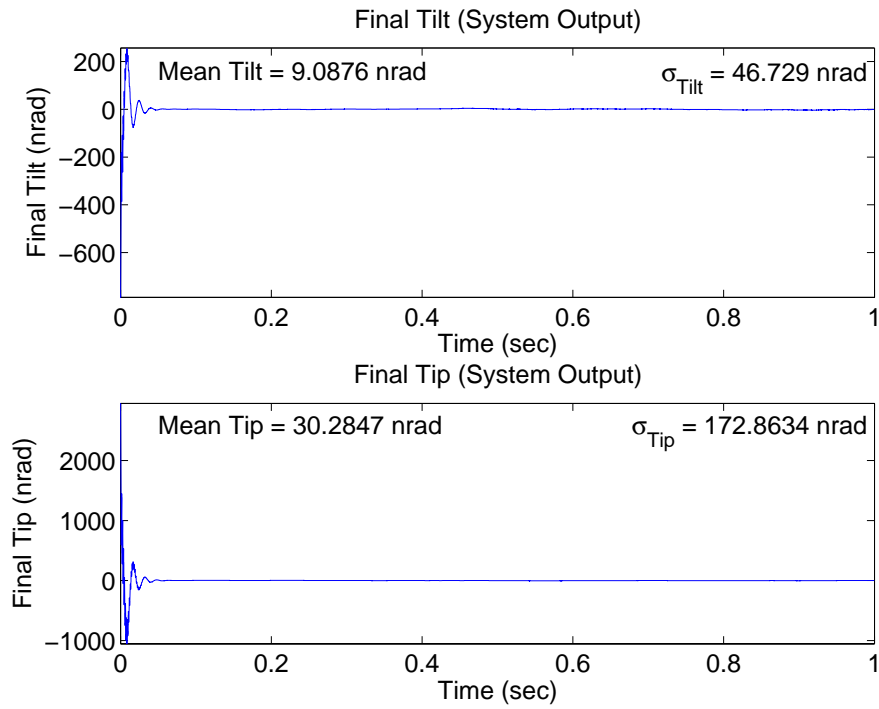


Figure 4.87 Case (14c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

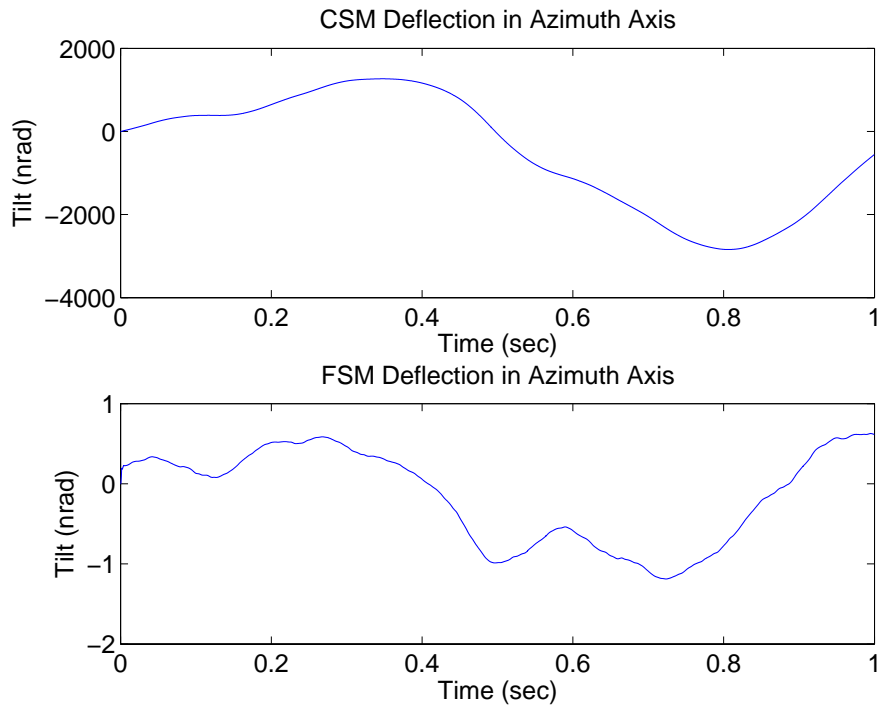


Figure 4.88 Case (1a): Steering Mirror Outputs (Azimuth Axis)

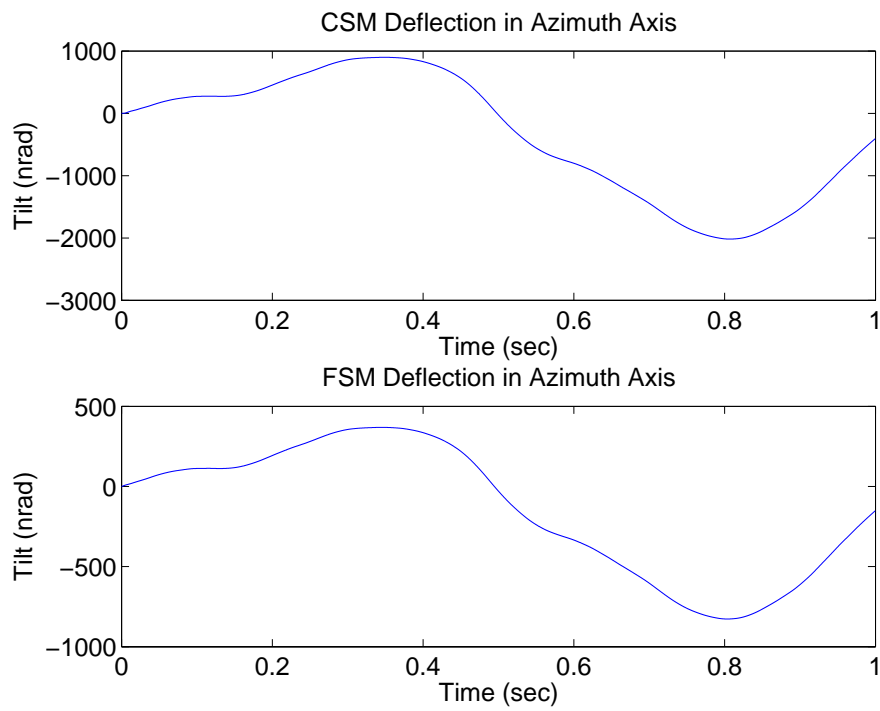


Figure 4.89 Case (1b): Steering Mirror Outputs (Azimuth Axis)

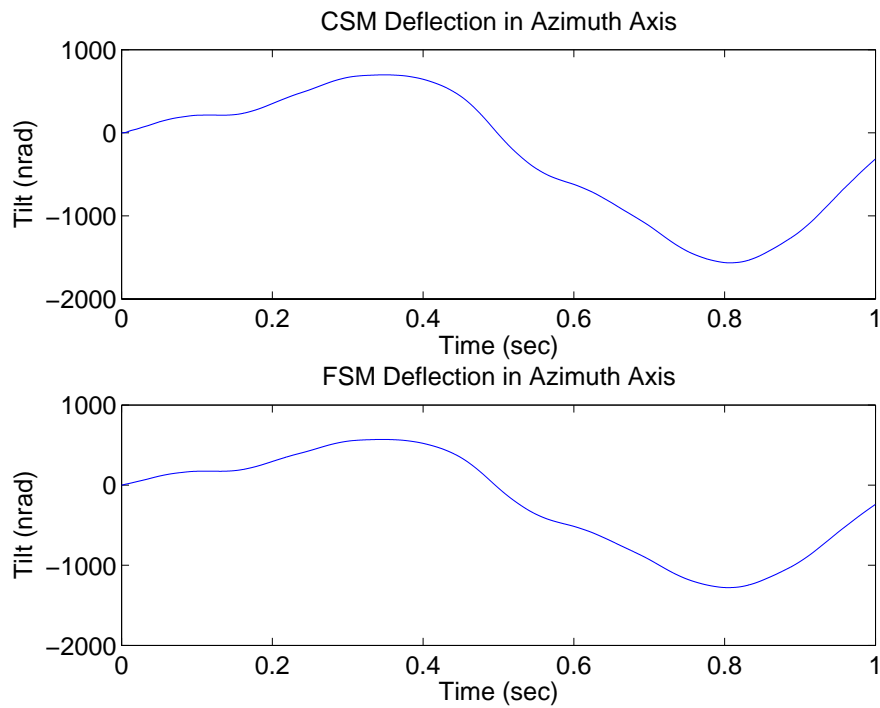


Figure 4.90 Case (1c): Steering Mirror Outputs (Azimuth Axis)

4.2.1.9 Summary of Baseline Results. Under nominal conditions of no time delays and no saturation, the LQG design in subcase (a) exhibits less transient amplitude, significantly less transient time, and better tilt/tip regulation performance than the designs of subcases (b) and (c). However, subject to delay times of three sample periods and more, the designs of subcases (b) and (c) are much more robust than subcase (a). The LQG design of subcase (c) consistently results in better regulation performance than subcase (b). Simulations with sensor saturation-only show that the transient time of the control system slightly increases if the tilt/tip disturbance is much larger than the saturation limits; however, performance is not affected once good tracking is established. Time delays greater than one sample period cause a much slower regulation response from the control system, resulting in degraded performance. Time delay response characteristics include a longer transient time, larger transient amplitude, and much more oscillation. Performance from subcases (c) and (b) remains stable up to and including a nine-sample time delay and eight-sample time delay, respectively, while performance from subcase (a) remains stable up to and including a three-sample time delay. The inclusion of pyramid sensor saturation with time delays of three sample periods and more actually resulted in improved responses (to disturbances much larger than the saturation limits) with much less transient oscillation and amplitude, as if the sensor saturation acts like a low-pass filter. The application of the LTR technique to each LQG design added robustness to one additional level of sample time delay, and performance of the LQG/LTR designs under nominal conditions did not significantly degrade.

4.2.2 Sensor Noise Cases. The purpose of Cases (15) to (26) is to evaluate the performance and robustness of the control system using noise-corrupted measurements of the tilt and tip disturbances. The pyramid sensor noise used in the simulations is described by Table 4.3. Cases (17) to (20) and (23) to (25) consist of sensor saturation and time delays in addition to sensor noise. Due to the minimal affect of saturation on the control system performance, as observed in the baseline cases, saturation is included in all simulations involving system time delays. Saturation-only and delay-time-only simulations are not conducted. The tilt and tip values used in Cases (15) to (22), and (26), are shown in Figures 4.3 and 4.15 based on the nominal Greenwood frequency of 86 Hz. Case (23) is

a 4 second simulation using tilt and tip values shown in Figures 4.4 and 4.16 based on a Greenwood frequency of 86 Hz. Case (24) is a 4 second simulation using tilt and tip values shown in Figures 4.5 and 4.17 based on a Greenwood frequency of 189.2 Hz, and Case (25) is a 4 second simulation using tilt and tip values shown in Figures 4.2 and 4.14 based on a Greenwood frequency of 43 Hz. The controller designs evaluated are based on the same three cost functions presented in the baseline case.

4.2.2.1 Cases (15) and (16). The \mathbf{Q} and \mathbf{R} matrices were tuned starting from the baseline values for each cost function based on performance in Cases (15) and (16). The tuning process consisted of determining the \mathbf{R} matrix values that provided best performance, followed by observing whether changes in the \mathbf{Q} matrix values had any further impact. Since the \mathbf{R} matrix values model the sensor noise, performance improved as the diagonal values were increased, and the off-diagonal values remained at zero. Table 4.7 shows the \mathbf{R}_{diag} values for each subcase. The second \mathbf{R}_{diag} value in each subcase,

Table 4.7 Kalman Filter Measurement Noise

Subcase	\mathbf{R}_{diag}
(a)	$\begin{bmatrix} 4E5 & 6E5 \end{bmatrix}$
(b)	$\begin{bmatrix} 9E4 & 2E5 \end{bmatrix}$
(c)	$\begin{bmatrix} 9E4 & 2E5 \end{bmatrix}$

corresponding to the elevation axis, is larger than the first since the actual elevation axis noise of the pyramid sensor is greater than the noise in the azimuth axis (as shown in Section 4.1.2.2). Increasing the \mathbf{R}_{diag} values beyond those shown in Table 4.7 causes the performance to worsen. To tune the \mathbf{Q} matrix, the off-diagonal entries were kept as zero while the diagonal entries were increased. As in the baseline case, performance degrades when the disturbance \mathbf{Q}_{diag} values are changed from the initial design values. Increasing the first six values of \mathbf{Q}_{diag} , which correspond to the CSM azimuth and elevation models, degrades performance while decreasing the values has no impact. Increasing the values corresponding to the FSM models, however, improved performance. Therefore, the \mathbf{Q}_{diag} values shown in Table 4.8 are used, where columns 1 to 6 correspond to the CSM model

states, columns 7 to 27 correspond to the FSM model states, and columns 28 and 29 are the WGN covariance values from the disturbance model developed in Section 3.3.2.2.

Table 4.8 Kalman Filter Dynamics Noise (Col. = Column)

	Q_{diag}		
Subcase	Col. 1 to 6	Col. 7 to 27	Col. 28 and 29
(a)	1	$1E7$	$2.7535E7$
(b)	1	$1E6$	$2.7535E7$
(c)	1	$1E6$	$2.7535E7$

The purpose of Cases (15) and (16) is to evaluate the performance of the control system using noisy measurements from the pyramid sensor. Figures 4.91 to 4.93 show the non-acquisition final mean absolute tilt/tip and standard deviation values, and Figures 4.94 to 4.96 show the acquisition final mean absolute tilt/tip and standard deviation values. As expected, the mean tilt and tip values in Case (15) increase from the values observed in baseline Case (1) due to the added measurement noise. In Case (15a), the mean values increase by as much as 20 nrad in tilt and 27 nrad in tip. Unlike the trend in baseline Case (1), Cases (15b) and (15c) outperform Case (15a) with better mean and standard deviation values. In Cases (15b) and (15c), the mean values increase from the Case (1) values by 13 nrad in tilt and 18 nrad in tip. Cases (15b) and (15c) show very similar performance, comparatively, with slightly better mean and standard deviation values in Case (15c). In Cases (15) and (16), better performance is seen in the azimuth axis, primarily due to the higher level of sensor noise in the elevation axis that could not be overcome by tuning. Different performance in each axis is also attributed to different sets of tilt and tip data used. The mean values in Case (16) also increase from the values in baseline Case (2). Like the baseline acquisition cases, the important result of Case (16) is that the system is able to regulate very large initial disturbances. The performance trend in Case (15) is also observed in Case (16) in which Cases (16b) and (16c) outperform Case (16a) in terms of mean and standard deviation values. The transient periods in Case (16) have the same characteristics seen in Case (2). Figures 4.97 to 4.99 show the optimal control inputs provided to the steering mirrors from the controller for non-acquisition runs, and

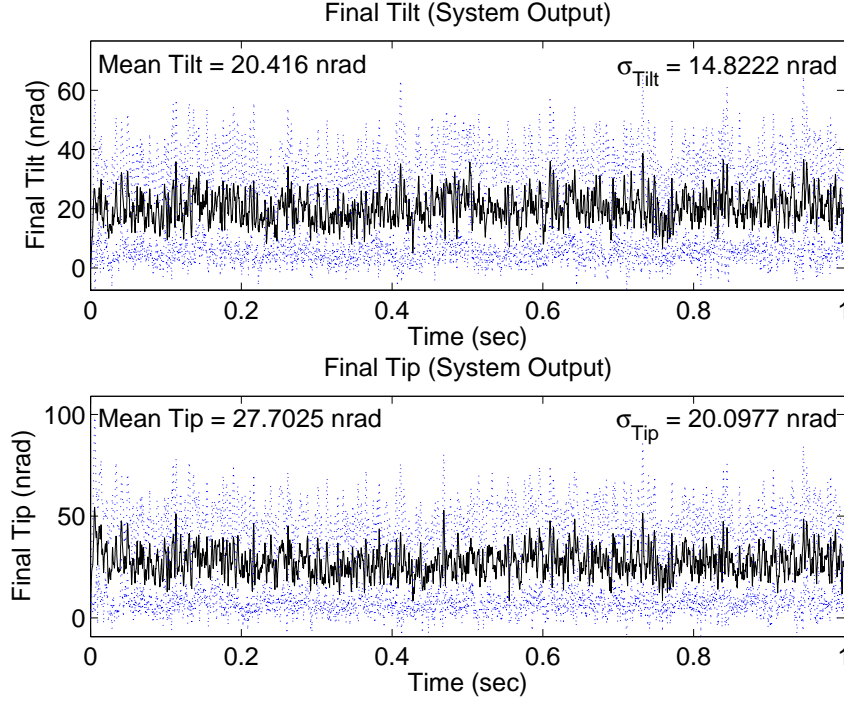


Figure 4.91 Case (15a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

Figures 4.100 to 4.102 show the control inputs for the acquisition cases. It is evident that the control system from subcase (a) does not combat the noise as well as subcases (b) and (c). The control inputs in Cases (15b), (15c), (16b), and (16c) are very smooth and appear much like the inputs from Cases (1) and (2), while the control inputs in Cases (15c) and (16c) are much noisier than in Cases (1) and (2). The final tilt and tip values in Cases (15b), and (15c) are also smoother than in Case (15c).

4.2.2.2 Cases (17) and (18). The purpose of Cases (17) and (18) is to evaluate the performance and robustness of the control system using saturated, noisy measurements from the pyramid sensor while subjected to various sample time delays.

One-Sample-Time Delay with Measurement Noise and Saturation.

Cases (17.1) and (18.1) simulate the control loop with a one-sample time delay and saturated, noisy measurements. Figures 4.103 to 4.105 show the non-acquisition final tilt/tip values, and Figures 4.106 to 4.108 show the acquisition final tilt/tip values. The same per-

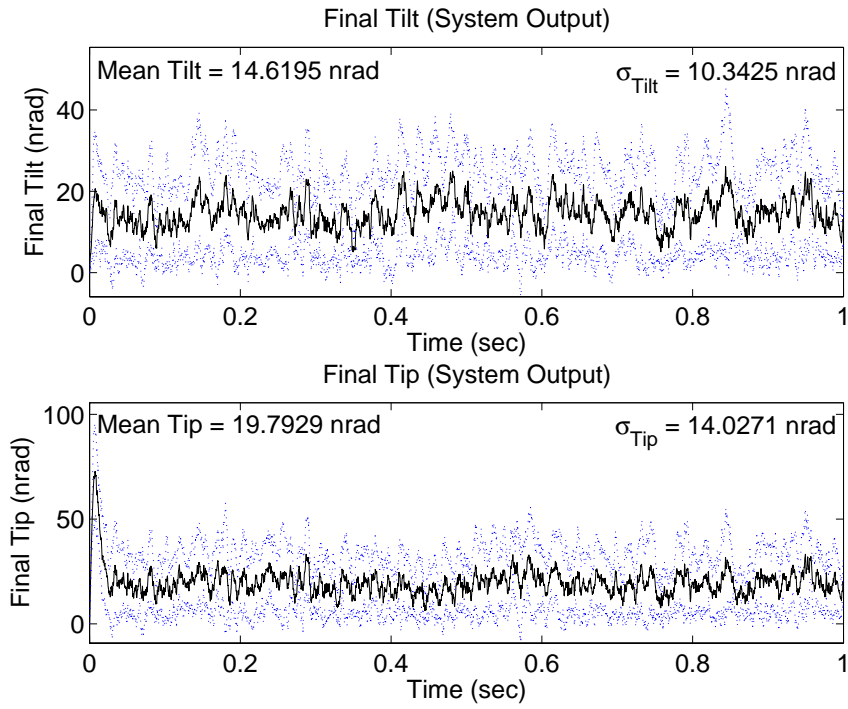


Figure 4.92 Case (15b): $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

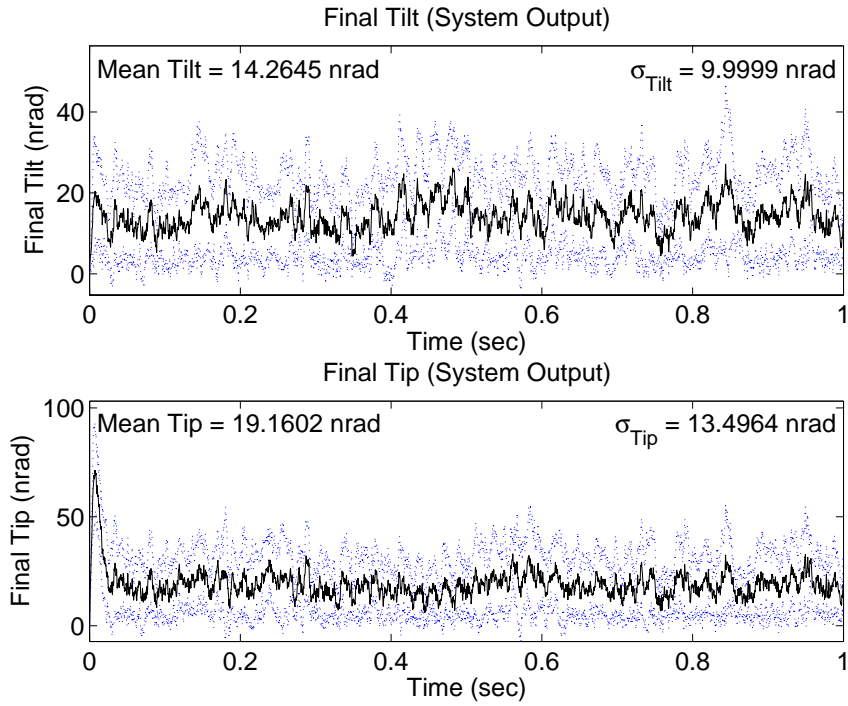


Figure 4.93 Case (15c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

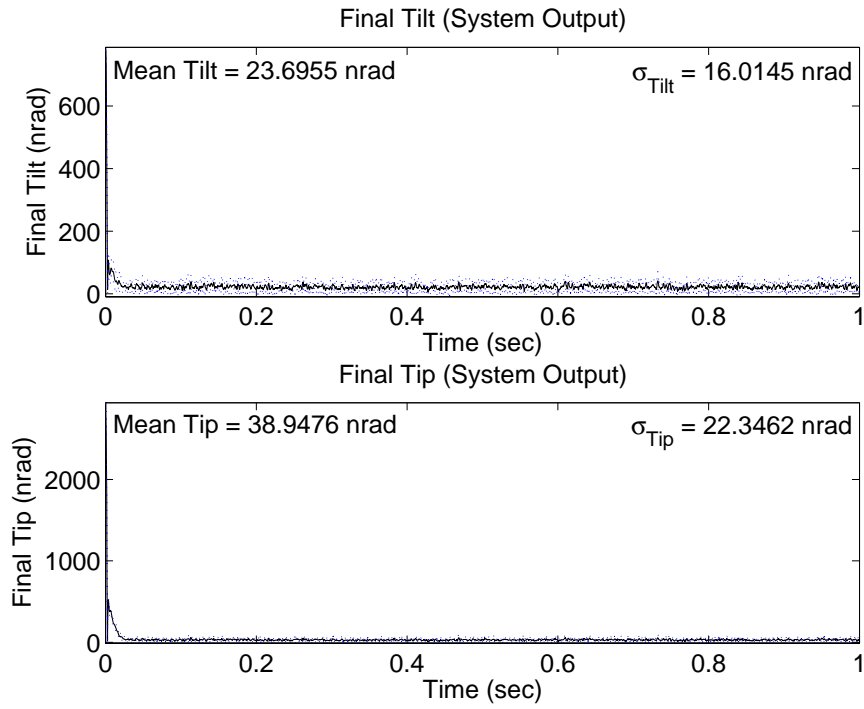


Figure 4.94 Case (16a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

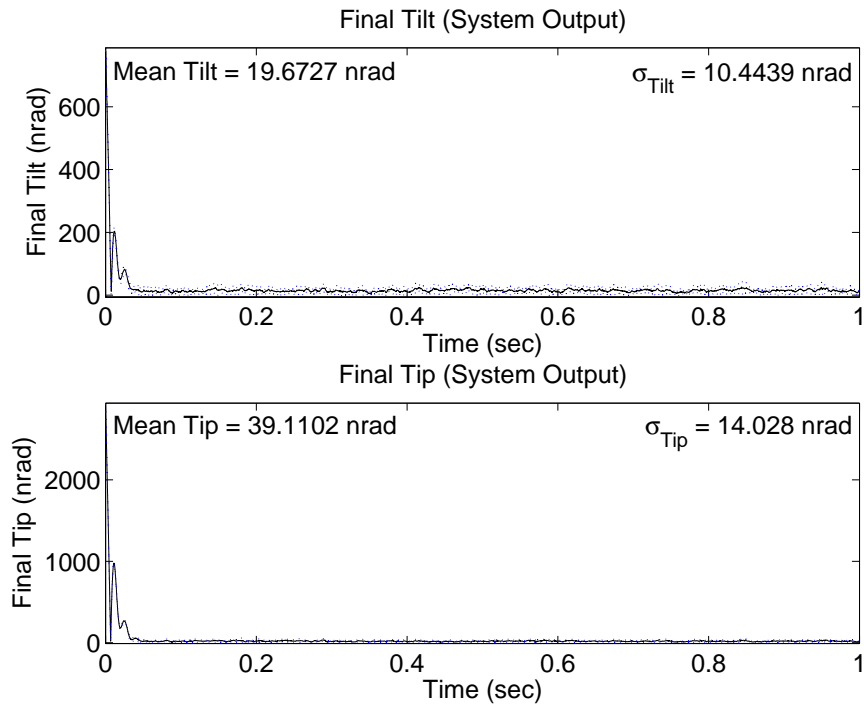


Figure 4.95 Case (16b): $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

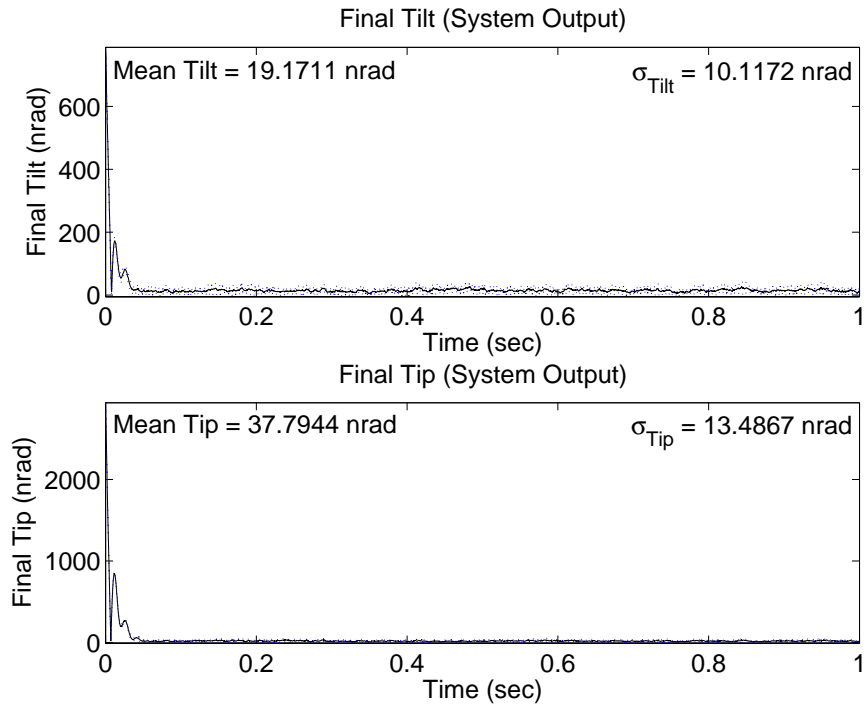


Figure 4.96 Case (16c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

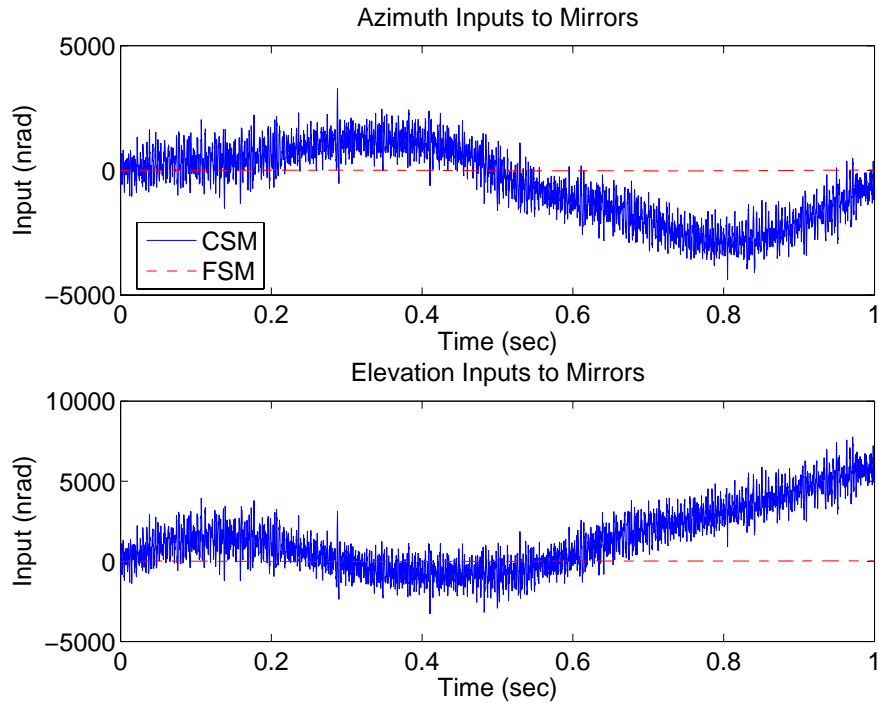


Figure 4.97 Case (15a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

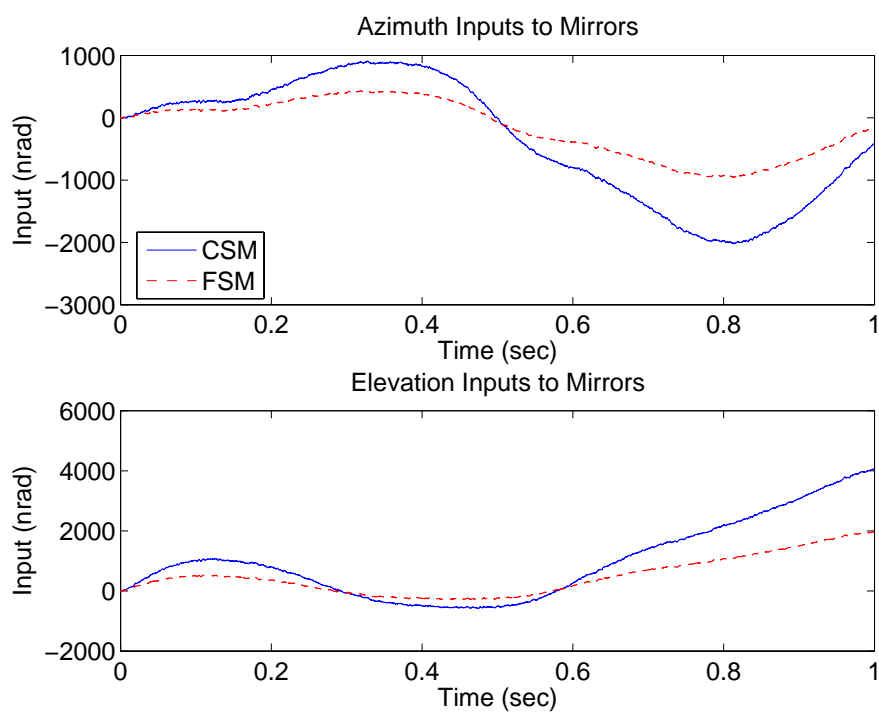


Figure 4.98 Case (15b): $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

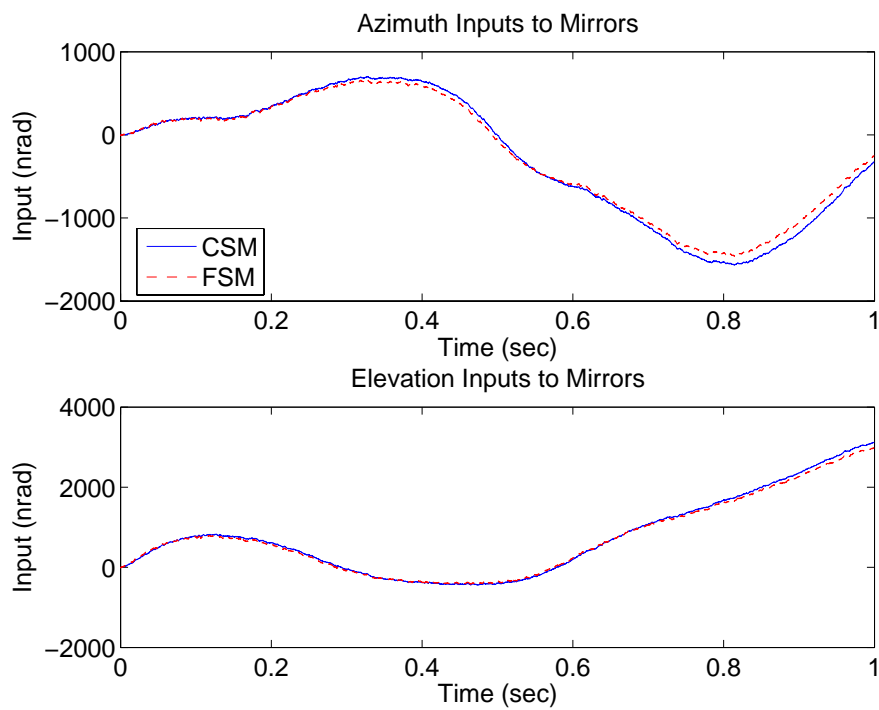


Figure 4.99 Case (15c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

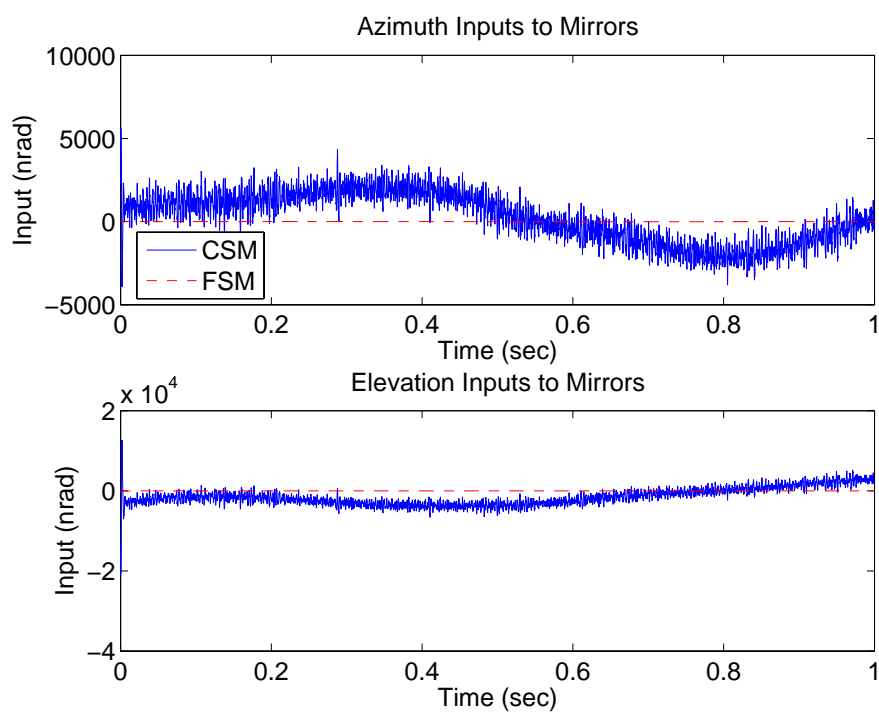


Figure 4.100 Case (16a): $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

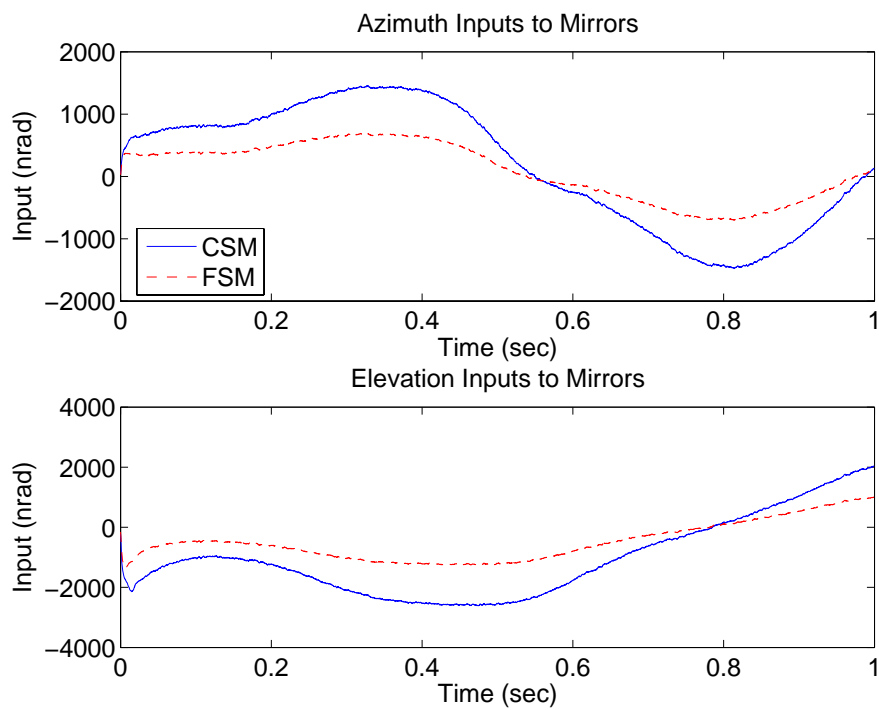


Figure 4.101 Case (16b): $\mathbf{U}_{diag} = [5E3 \quad 5E3 \quad 1E4 \quad 1E4]$

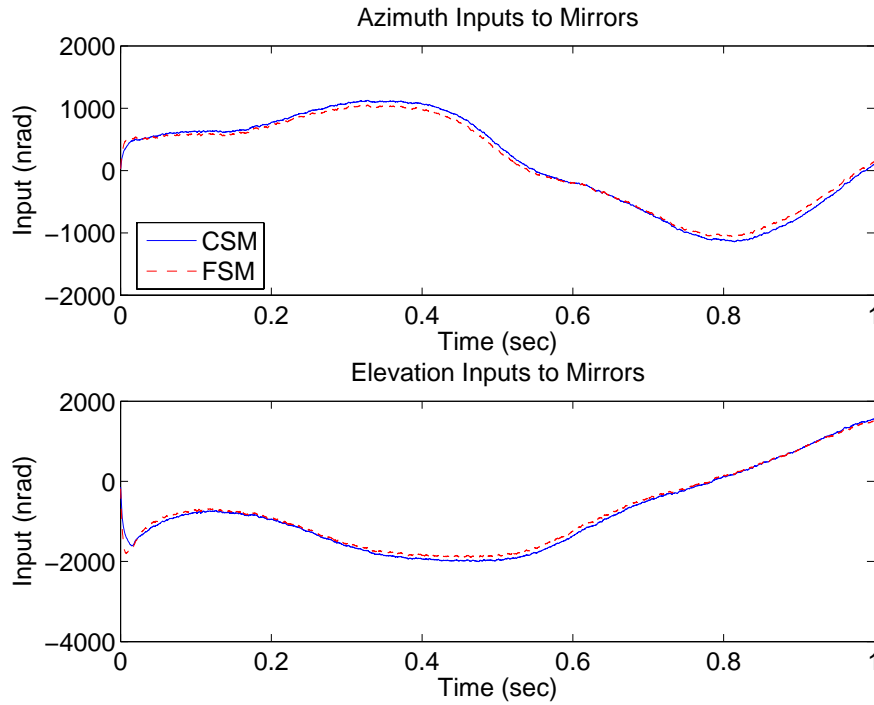


Figure 4.102 Case (16c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

formance trends from the baseline cases are seen in Cases (17.1) and (18.1) as compared to Cases (15) and (16), in which the non-acquisition performance remains almost identical while the acquisition performance worsens. The mean and standard deviation values in Case (17) increase slightly by no more than 1.4 nrad (Case (17.1a)); though the increase is small, the most increase is in Case (17.1a). The mean values in Case (18.1) increase more significantly due to an increase in the time it takes the control system to drive the initial tilt and tip to zero. The standard deviation values in Case (18.1a) decrease by no more than 0.8 nrad, while in Cases (18.1b) and (18.1c) the values slightly increase. Unlike Case (17), the mean values in Cases (18.1b) and (18.1c) increase more than in Case (18.1a) such that Case (18.1a) has the best mean value in the elevation axis, however, the standard deviation values in Cases (18.1b) and (18.1c) remain less than in Case (18.1a). The control inputs are not shown since there are no significant changes from Cases (15) and (16).

Five-Sample-Time Delay with Measurement Noise and Saturation.

Cases (17.5) and (18.5) simulate the control loop with a five-sample time delay and satu-

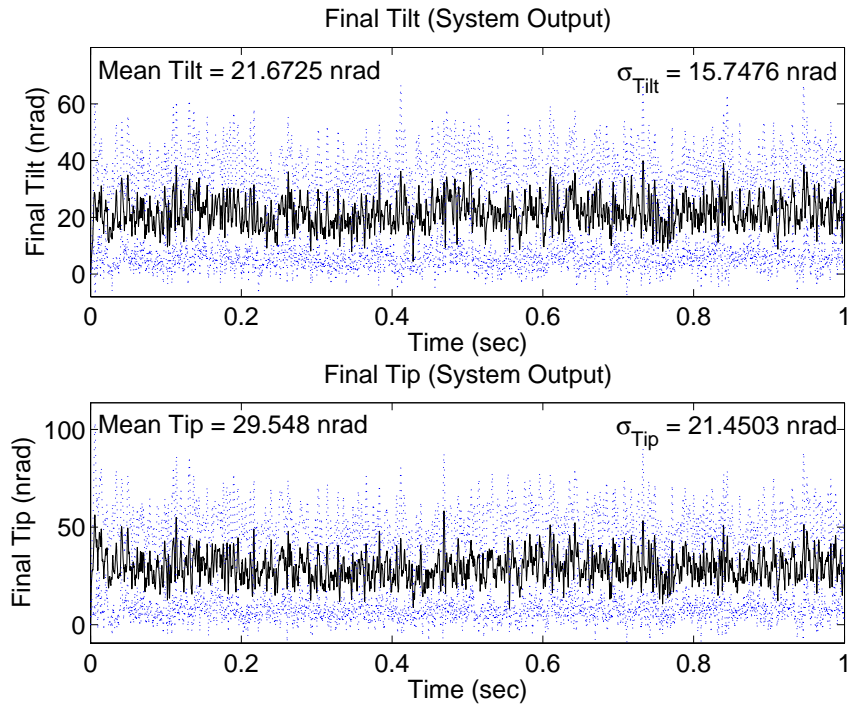


Figure 4.103 Case (17.1a): Delay 1, $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

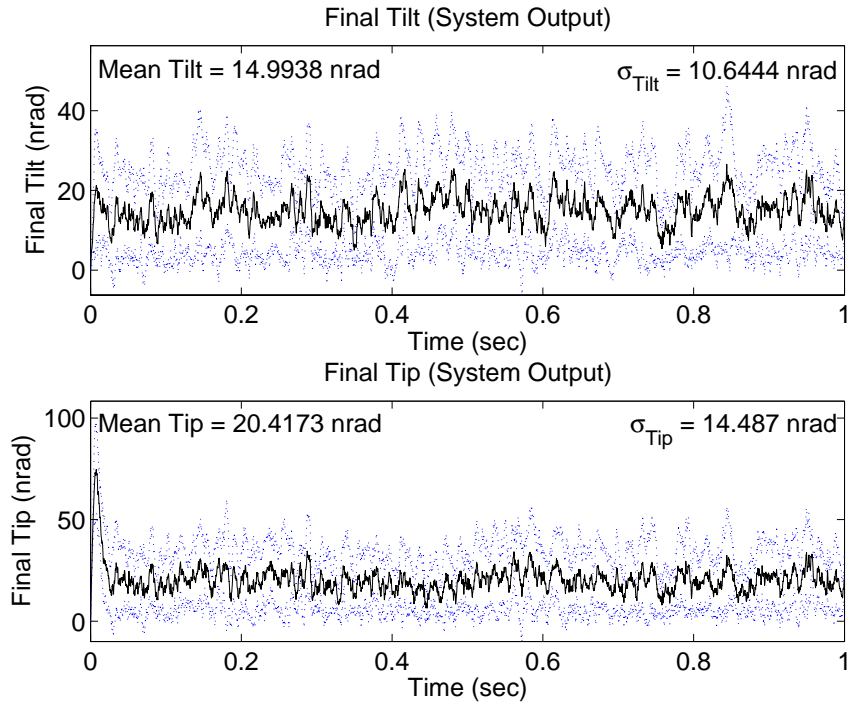


Figure 4.104 Case (17.1b): Delay 1, $\mathbf{U}_{diag} = [5E3 \quad 5E3 \quad 1E4 \quad 1E4]$

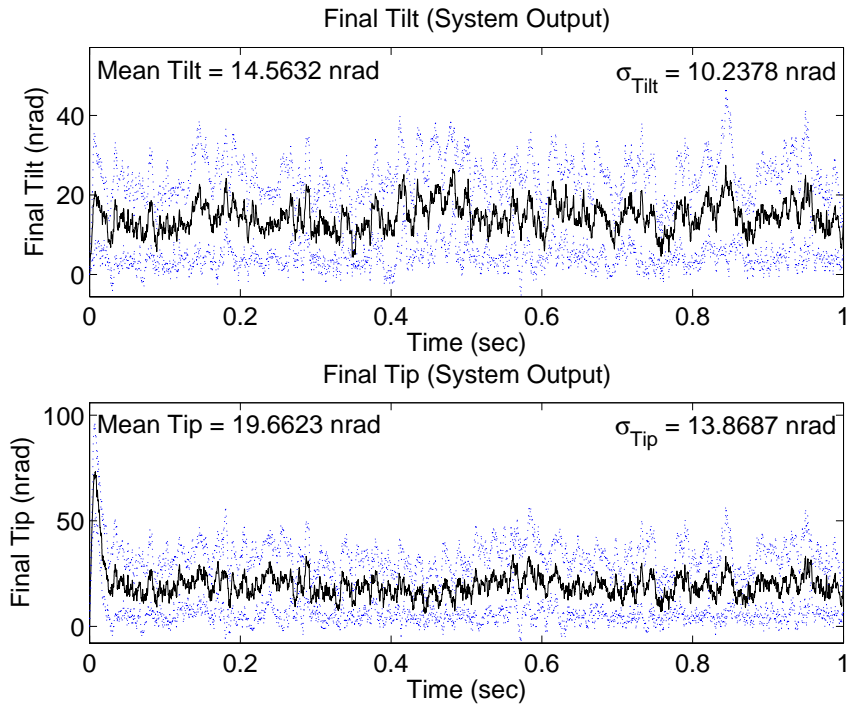


Figure 4.105 Case (17.1c): Delay 1, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

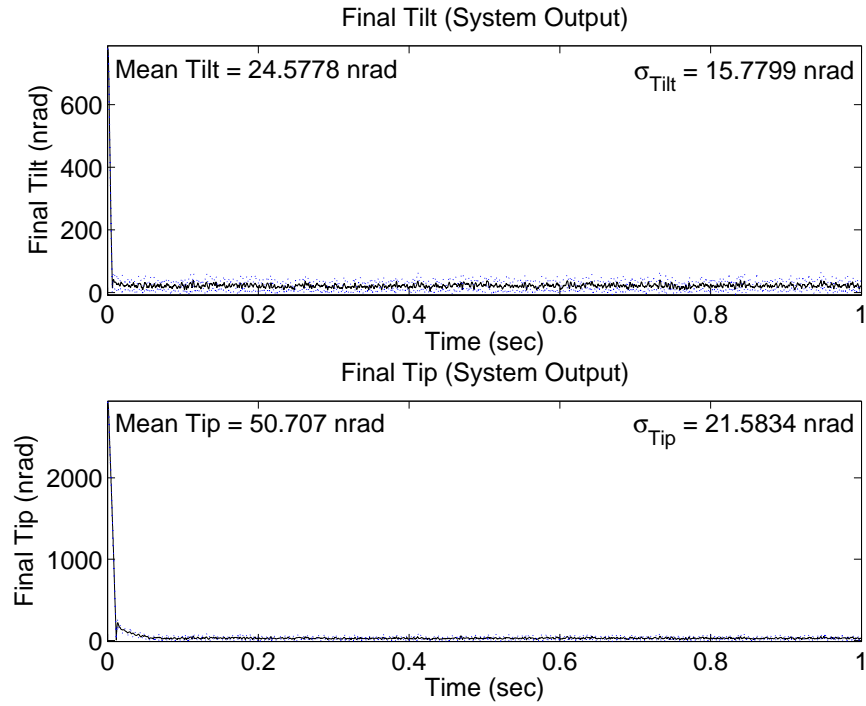


Figure 4.106 Case (18.1a): Delay 1, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

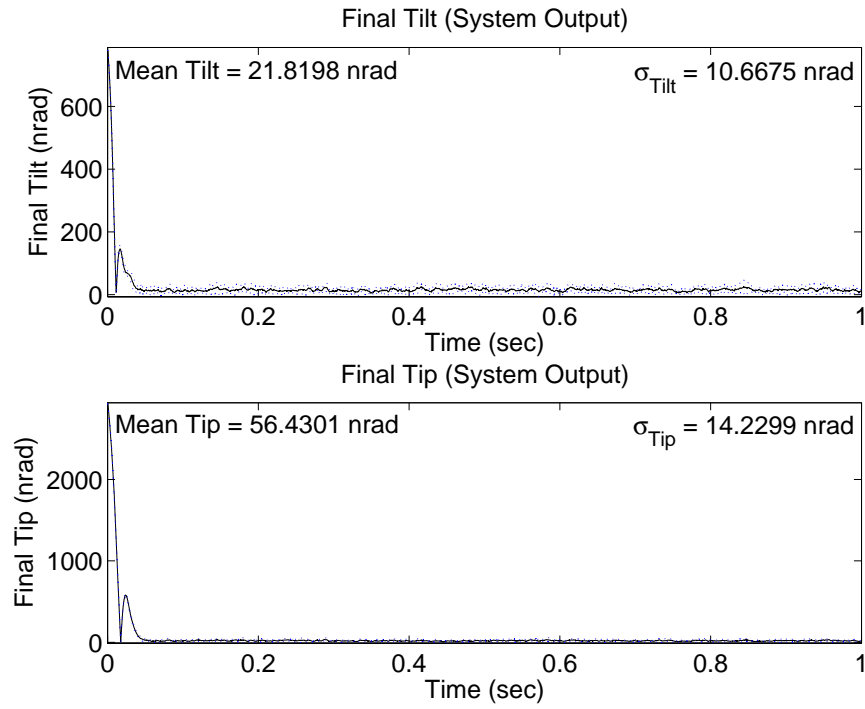


Figure 4.107 Case (18.1b): Delay 1, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

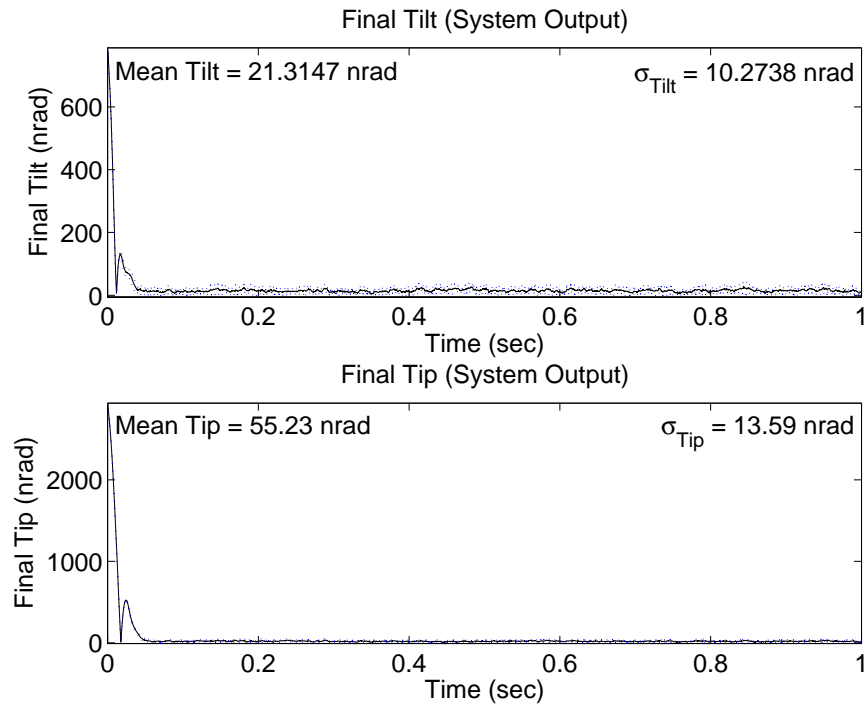


Figure 4.108 Case (18.1c): Delay 1, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

rated, noisy measurements. Figures 4.109 to 4.111 show the mean and standard deviation values for Case (17.5). Unlike Case (17.1), the mean values in Case (17.5) increase more significantly, especially Case (17.5a). In Case (17.5a), the mean tilt increases by 7 nrad, the mean tip increases by 10 nrad, and the standard deviation values increase by as much as 7 nrad. In Cases (17.5b) and (17.5c), the mean values increase no more than 3.4 nrad and the standard deviation values increase no more than 2.5 nrad. Case (17.5c) has slightly better performance than Case (17.5b). The most noteworthy result of Cases (17.5) and (18.5) is that stable performance is achieved in subcase (a), while in the baseline cases the results showed unstable performance with a four-sample delay (Case (7.4a)). This may be due to the different \mathbf{Q} and \mathbf{R} tuning parameters, or the sensor noise is contributing constructively to the stability of the system, though less likely. Case (27) re-evaluates some of the baseline cases using the \mathbf{Q} and \mathbf{R} matrices tuned for the sensor noise cases to determine if the additional stability is due to the new tuning parameters. The control inputs of Cases (17.5) and (18.5) are identical to Cases (17.1) and (18.1), and the results of Case (18.5) show the same characteristics and trends as Case (18.1) in which the mean and standard deviation values, as well as the transient period, increase.

Ten-Sample-Time Delay with Measurement Noise and Saturation.

Cases (17.10) and (18.10) simulate the control loop with a ten-sample time delay and saturated, noisy measurements. Figures 4.112 to 4.114 display the performance in Case (17.10), while Figures 4.121 to 4.123 show the performance in Case (18.10). All subcases show stable performance despite results from the baseline cases which indicated unstable performance by all three controller designs without LTR. The general trend from Cases (17.5) and (18.5) is also seen in the following results. Cases (17.10c) and (18.10c) show the best performance while Cases (17.10a) and (18.10a) show significant performance degradation. As compared to Case (17.5a), the mean tilt and tip values in Case (17.10a) increase by 38 nrad and 75 nrad respectively and the tilt and tip standard deviation increases by as much as 25 nrad and 48 nrad respectively. The performance in Cases (17.10b) and (17.10c) also worsens, but not as severely as in Case (17.10a). The mean tilt and tip values in Case (17.10b) increase by 6 nrad and 9 nrad, respectively, and the standard deviation increases by no more than 6 nrad, while the mean tilt and tip values in Case (17.10c) increase by

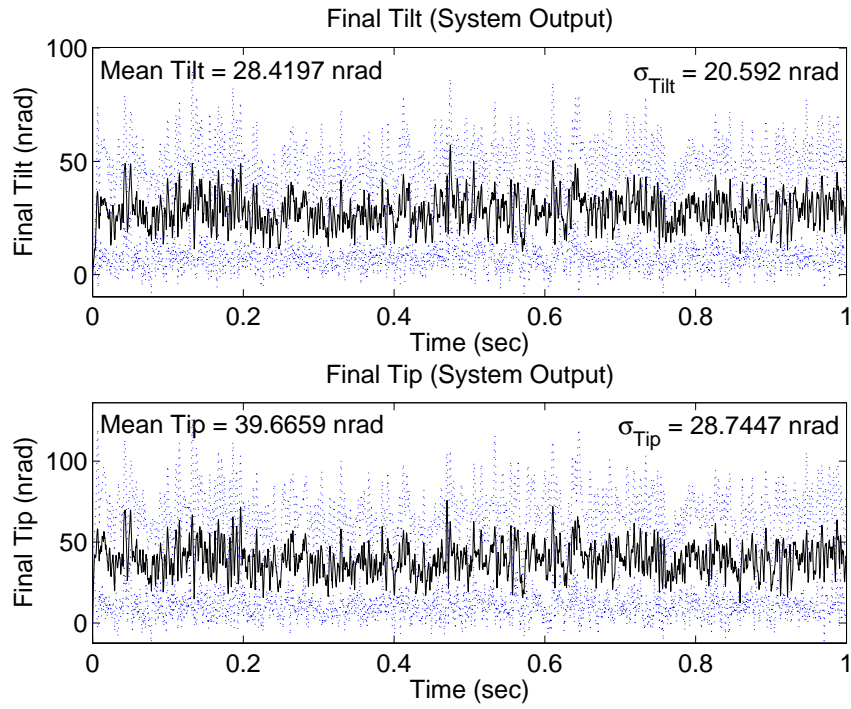


Figure 4.109 Case (17.5a): Delay 5, $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

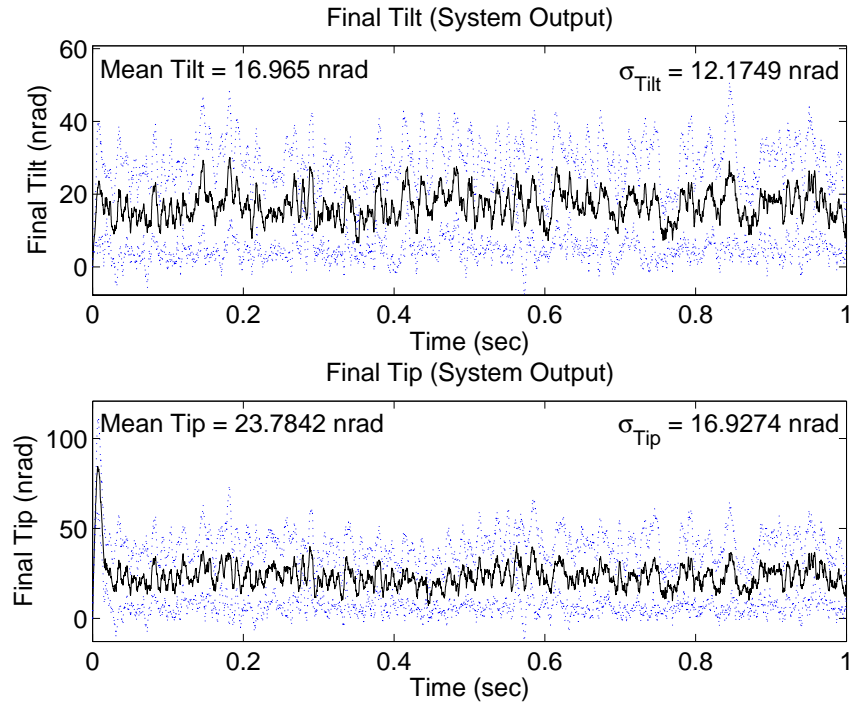


Figure 4.110 Case (17.5b): Delay 5, $\mathbf{U}_{diag} = [5E3 \quad 5E3 \quad 1E4 \quad 1E4]$

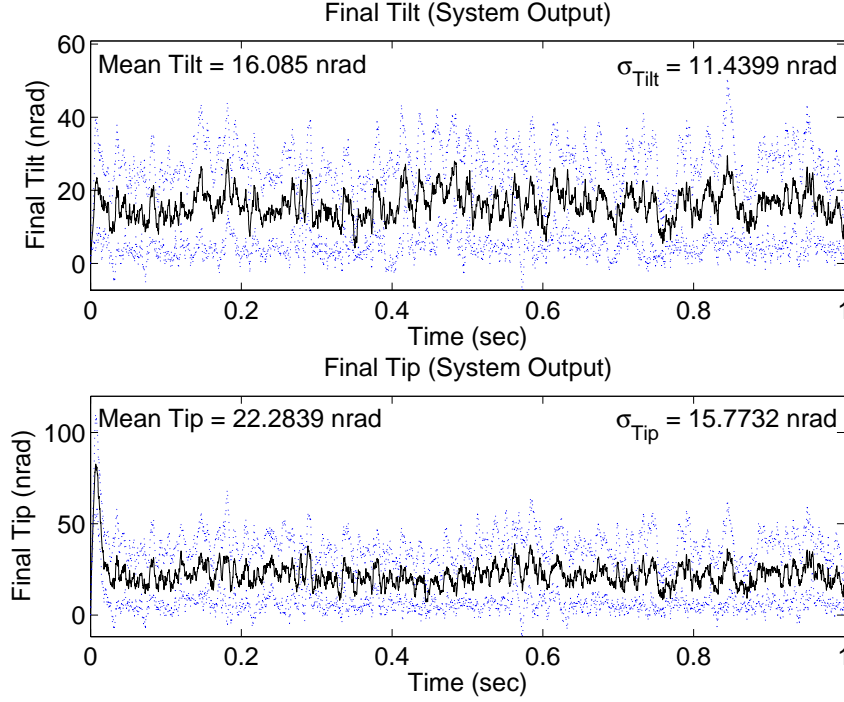


Figure 4.111 Case (17.5c): Delay 5, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

3.3 nrad and 6 nrad, respectively, and the standard deviation increases by no more than 5 nrad. Case (18.10) shows more oscillation in an extended transient period, which causes performance to degrade in all subcases, especially Case (18.10a), but the control system still maintains the ability to regulate the large initial disturbance despite the latency, saturation, and noise. The control inputs for Cases (17.10) and (18.10) are shown in Figures 4.118 to 4.123. The inputs maintain the same characteristics seen in all sensor noise cases except that the initial transient period is longer.

4.2.2.3 Cases (19) and (20). The purpose of Cases (19) and (20) is to evaluate the performance and robustness of the LQG/LTR designs, simulated with a ten-sample time delay, sensor saturation, and sensor noise, and compare the results to Cases (17.10) and (18.10). The objective is to determine if any benefit is gained from using the LTR technique since the original control systems remained stable through all sensor noise cases. The same LQG/LTR cost functions used in the baseline cases are simulated in Cases (19) and (20). The mean tilt/tip and standard deviation values for Case (19.10) are

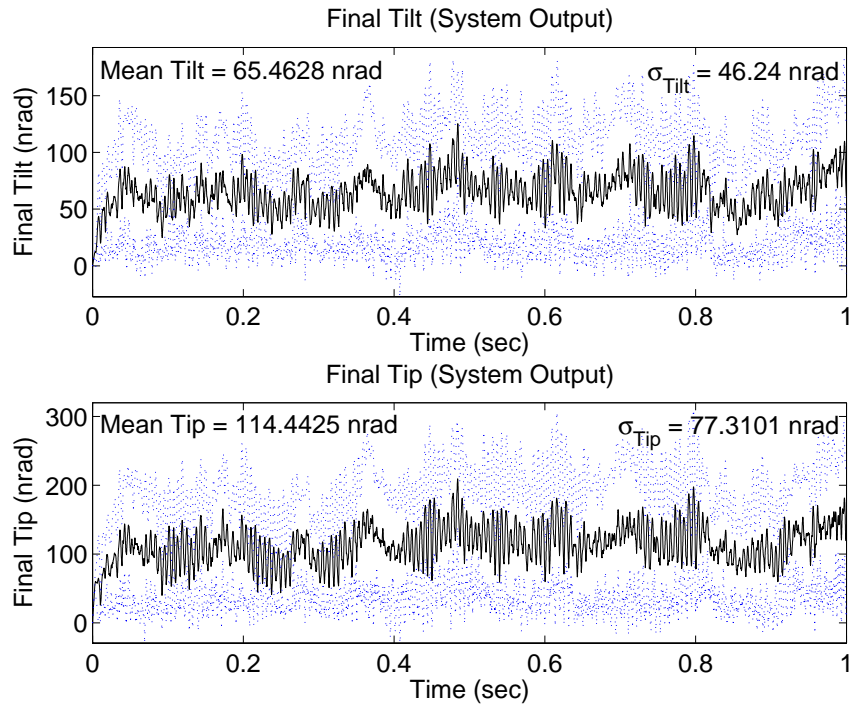


Figure 4.112 Case (17.10a): Delay 10, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

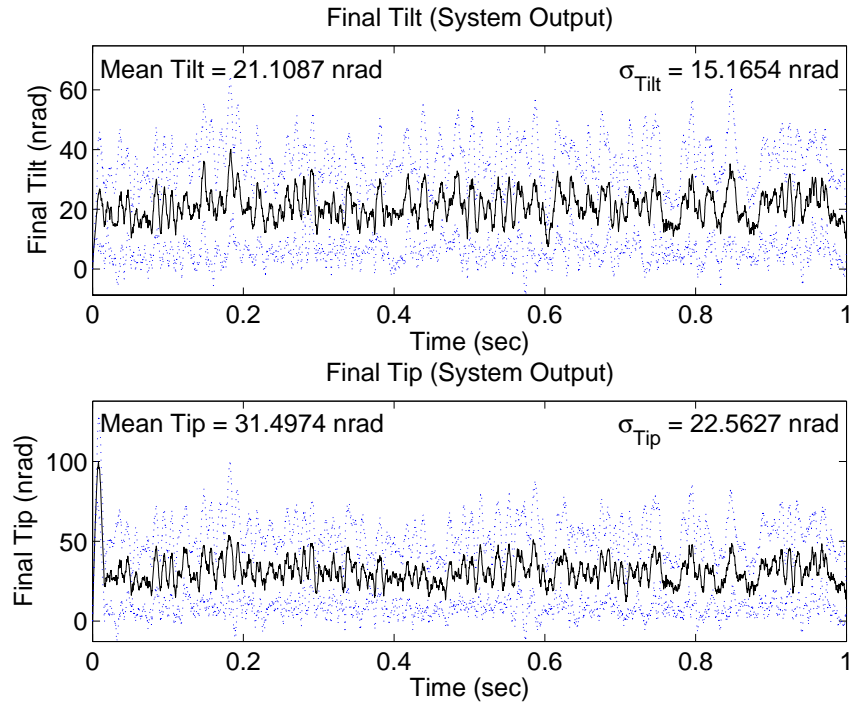


Figure 4.113 Case (17.10b): Delay 10, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

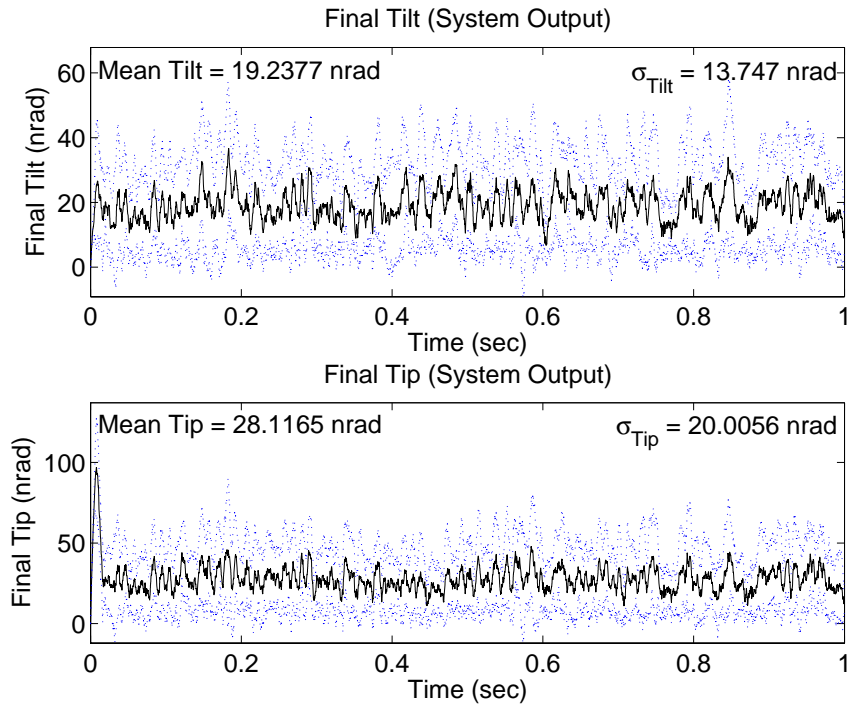


Figure 4.114 Case (17.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

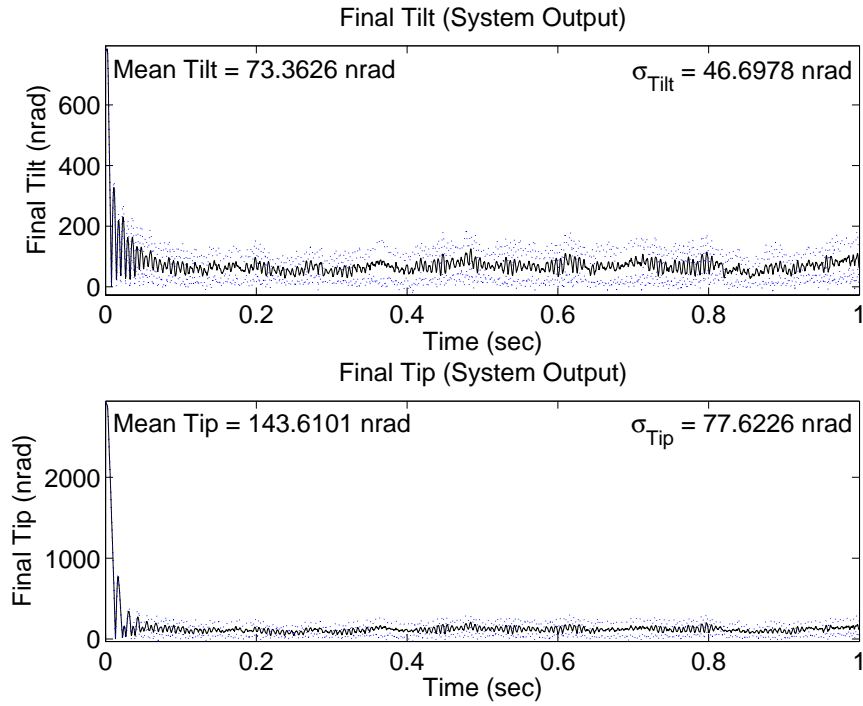


Figure 4.115 Case (18.10a): Delay 10, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

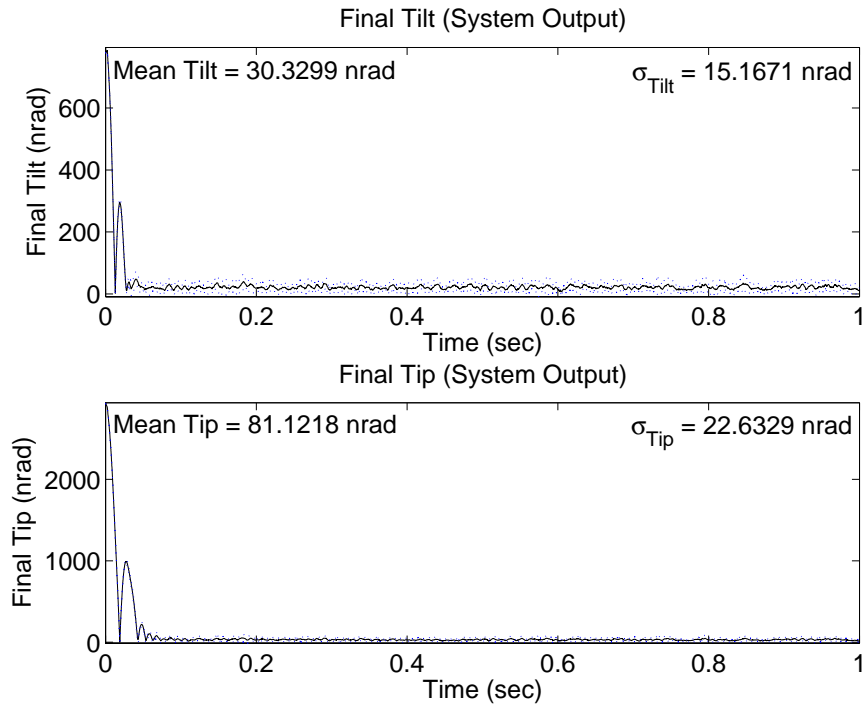


Figure 4.116 Case (18.10b): Delay 10, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

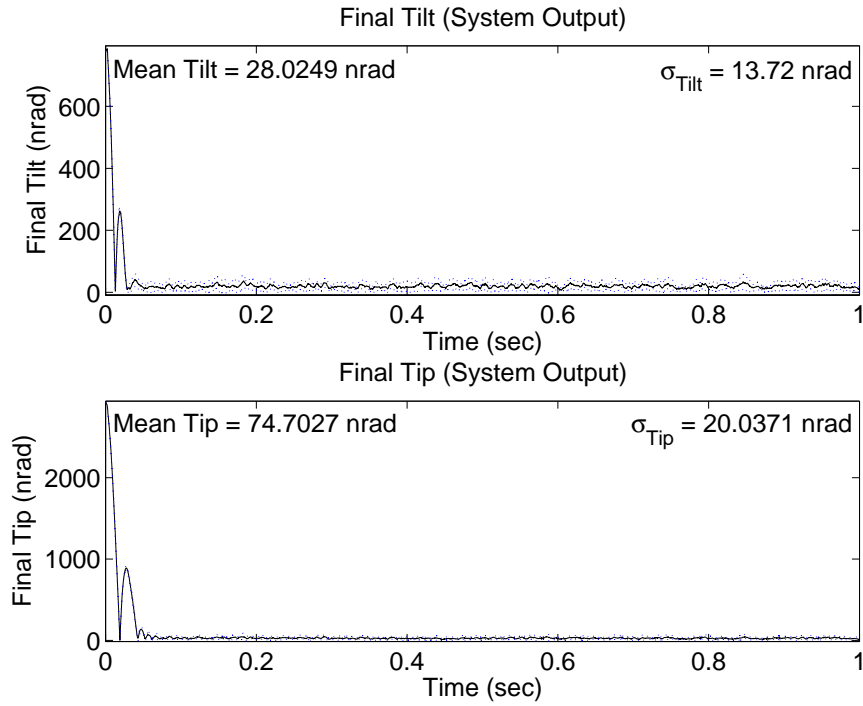


Figure 4.117 Case (18.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

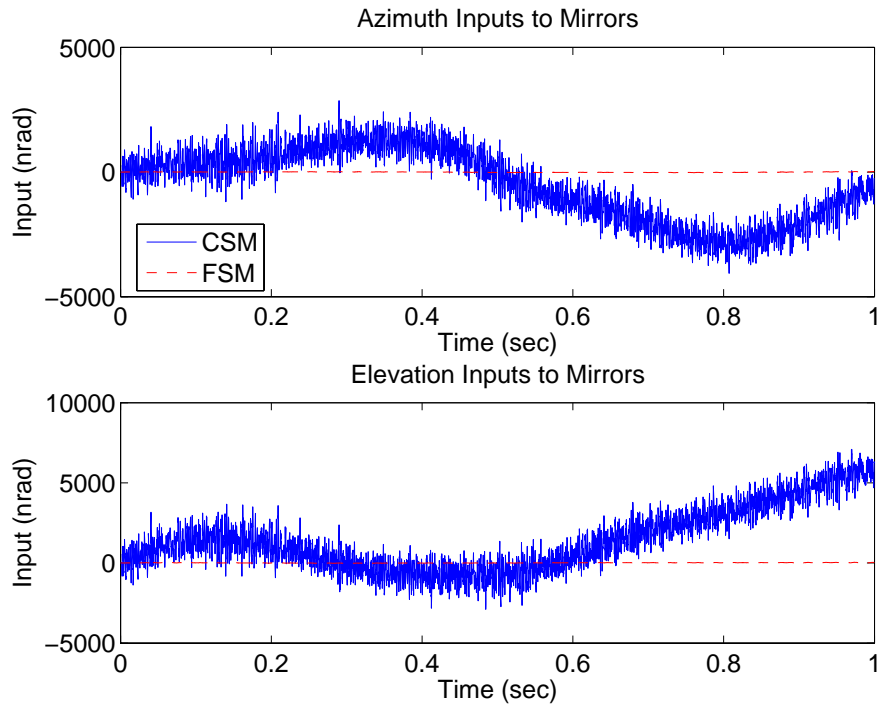


Figure 4.118 Case (17.10a): Delay 10, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

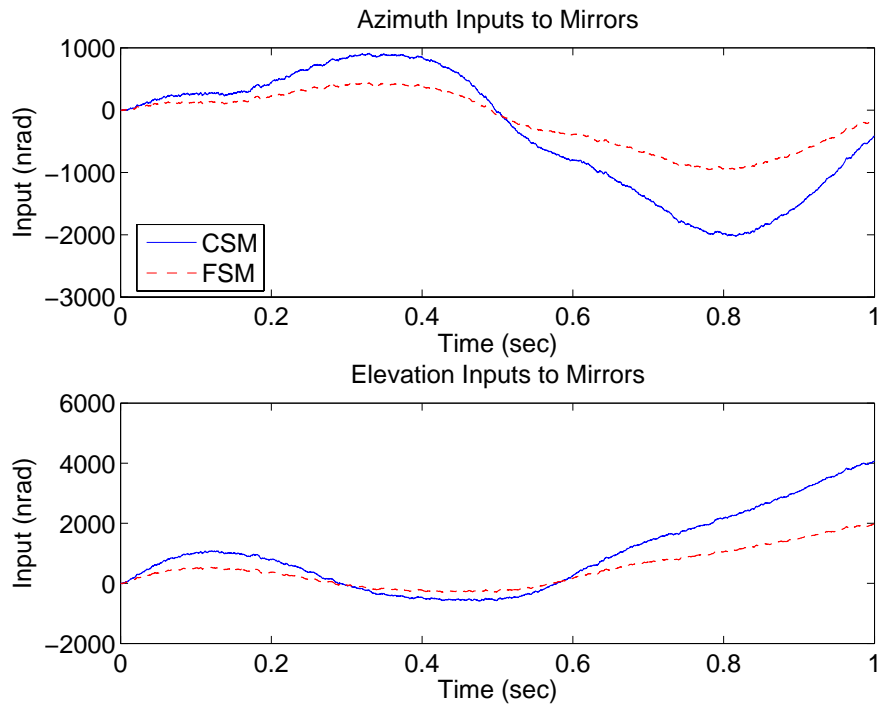


Figure 4.119 Case (17.10b): Delay 10, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

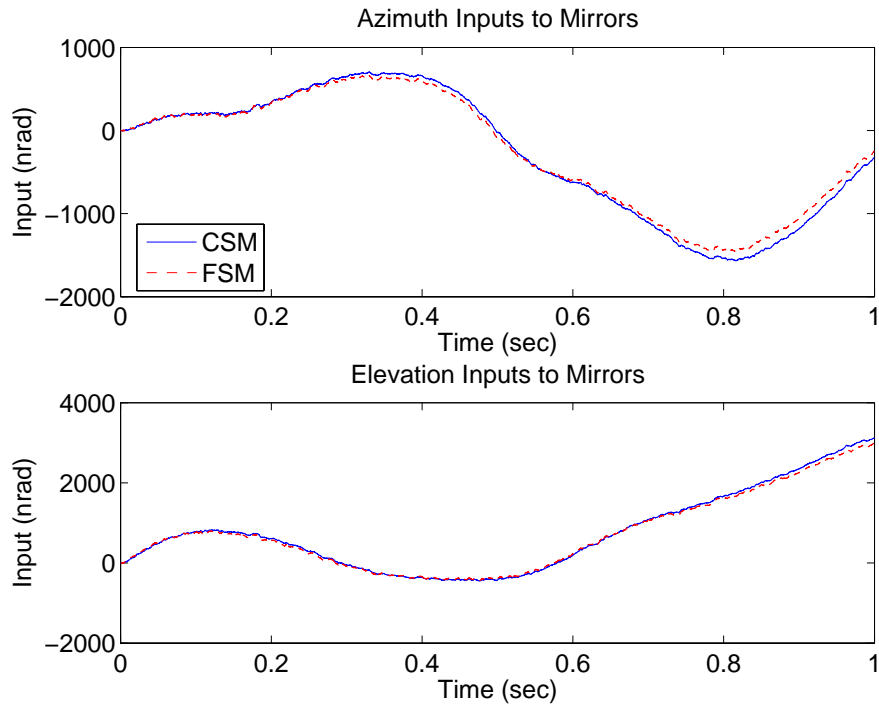


Figure 4.120 Case (17.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

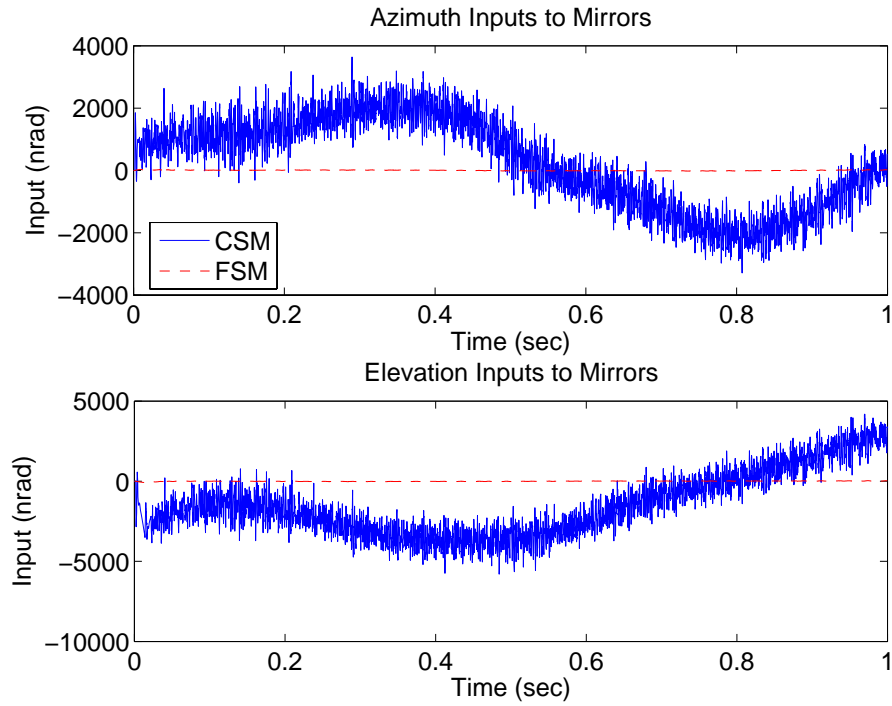


Figure 4.121 Case (18.10a): Delay 10, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

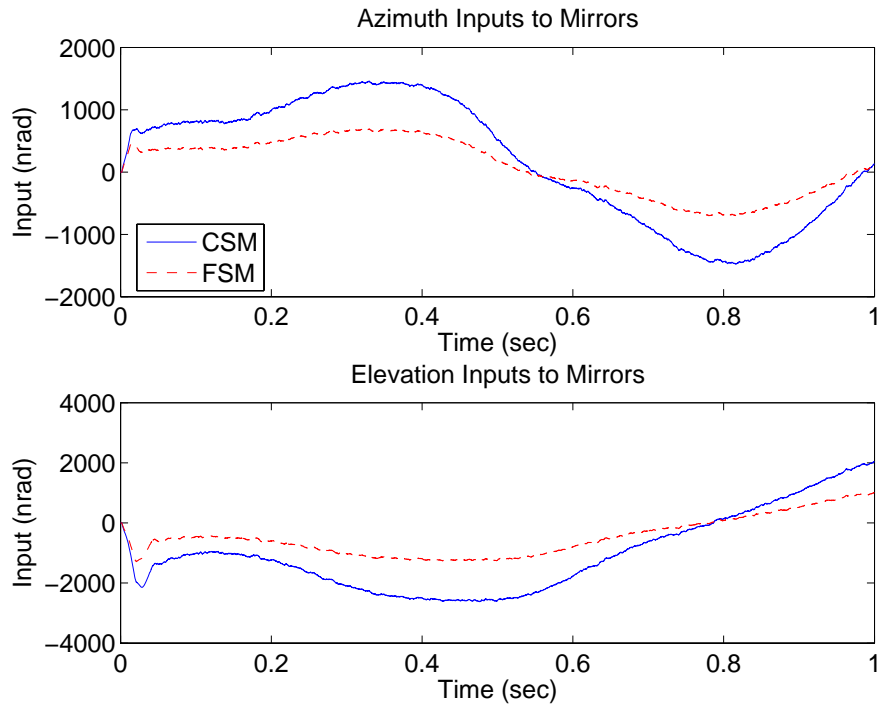


Figure 4.122 Case (18.10b): Delay 10, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

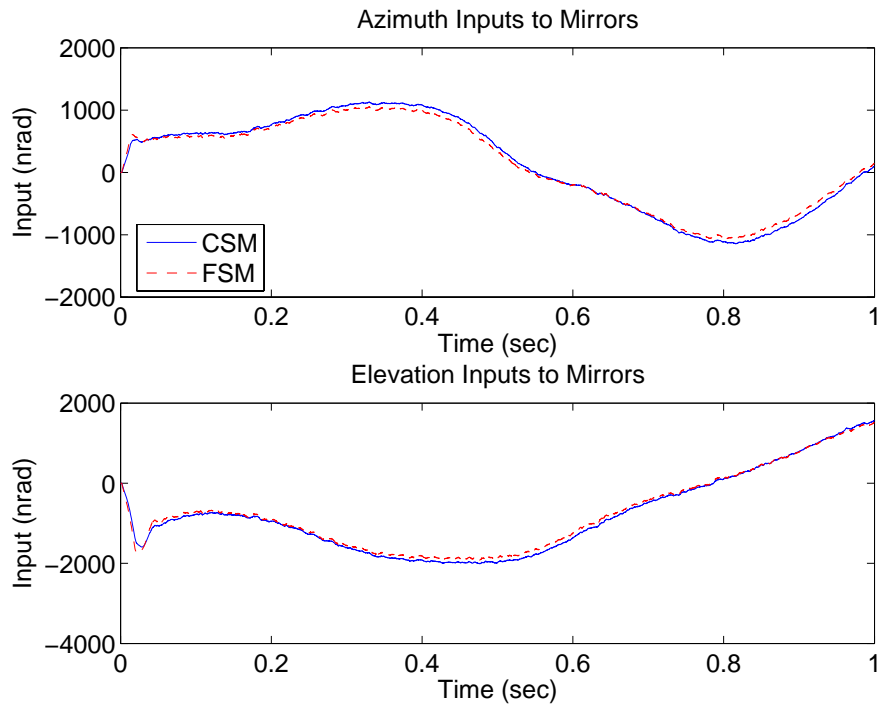


Figure 4.123 Case (18.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

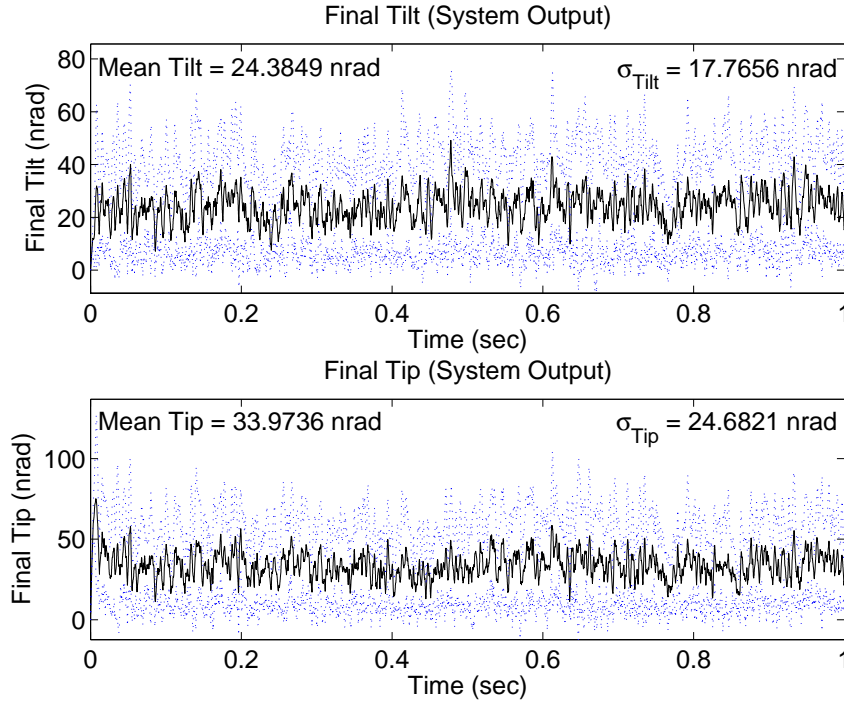


Figure 4.124 Case (19.10a): Delay 10, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

shown in Figures 4.124 to 4.126. The performance in Case (19.10a) improves significantly compared to Case (17.10a). In Case (19.10a), the mean tilt and tip values decrease by 40 nrad and 80 nrad, respectively, and the tilt and tip standard deviation decreases by 28 nrad and 52 nrad, respectively. The high frequency oscillation present in the tilt and tip values throughout the entire Case (17.10a) simulation is removed in Case (19.10a). The mean tilt/tip and standard deviation values in Cases (17.10b) and (17.10c) slightly decrease by 2 nrad at most. Figures 4.127 to 4.129 show the mean and standard deviation values for Case (20.10). The same trend in Case (19.10) is evident in Case (20.10). The transient oscillations in Case (18.10a) are no longer present in Case (20.10a), and the transient characteristics of Cases (20.10b) and (20.10c) do not change from Case (18). The control inputs are identical to Cases (17) and (18), therefore, are not displayed. Subcases (b) and (c) still outperform subcase (a).

4.2.2.4 Cases (21) and (22). The purpose of Cases (21) and (22) is to evaluate the performance of the LQG/LTR designs under the minimal condition of sensor

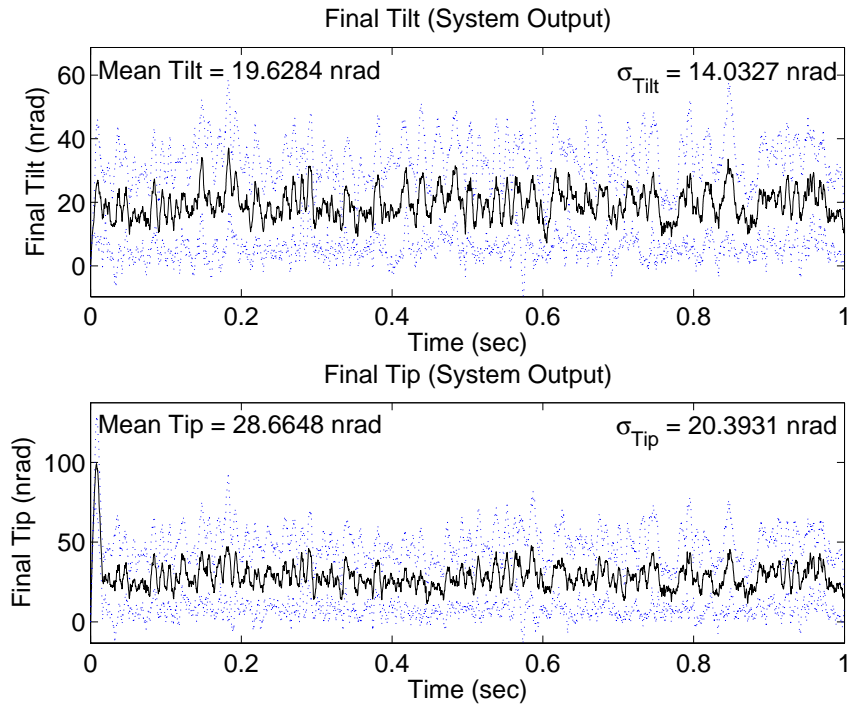


Figure 4.125 Case (19.10b): Delay 10, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

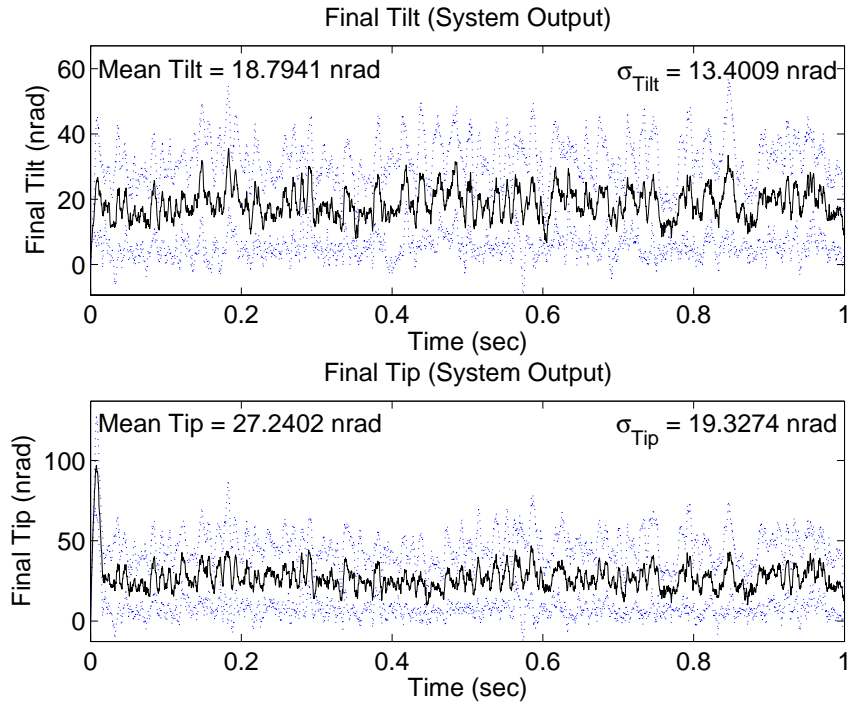


Figure 4.126 Case (19.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

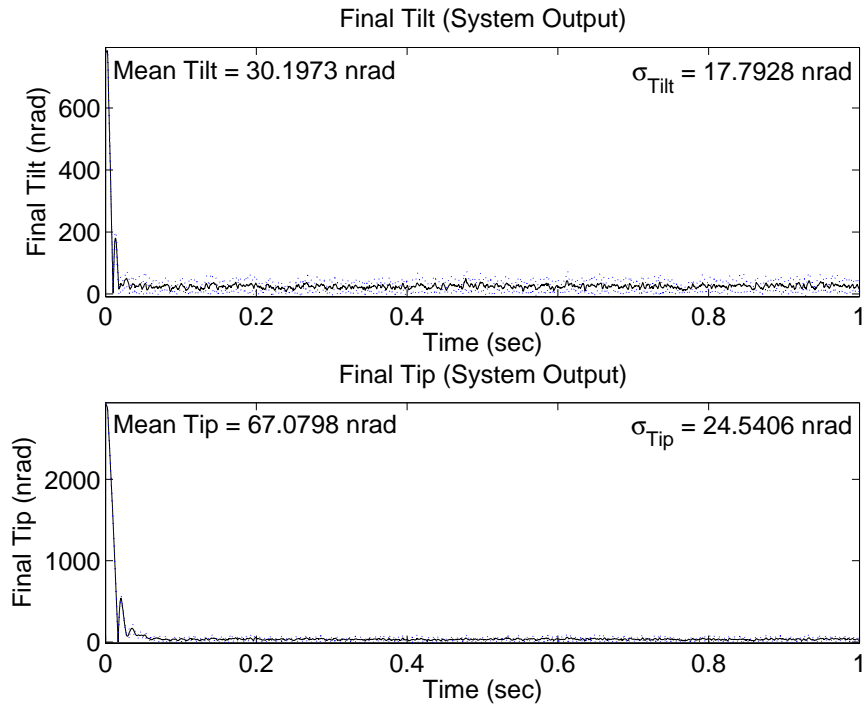


Figure 4.127 Case (20.10a): Delay 10, $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

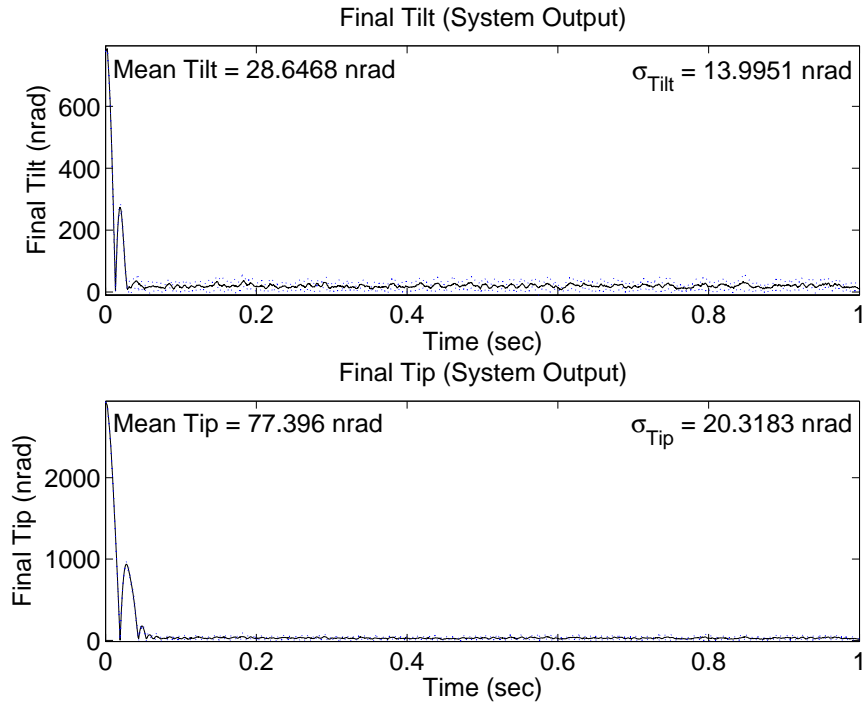


Figure 4.128 Case (20.10b): Delay 10, $\mathbf{U}_{diag} = [5E3 \quad 5E3 \quad 1E4 \quad 1E4]$

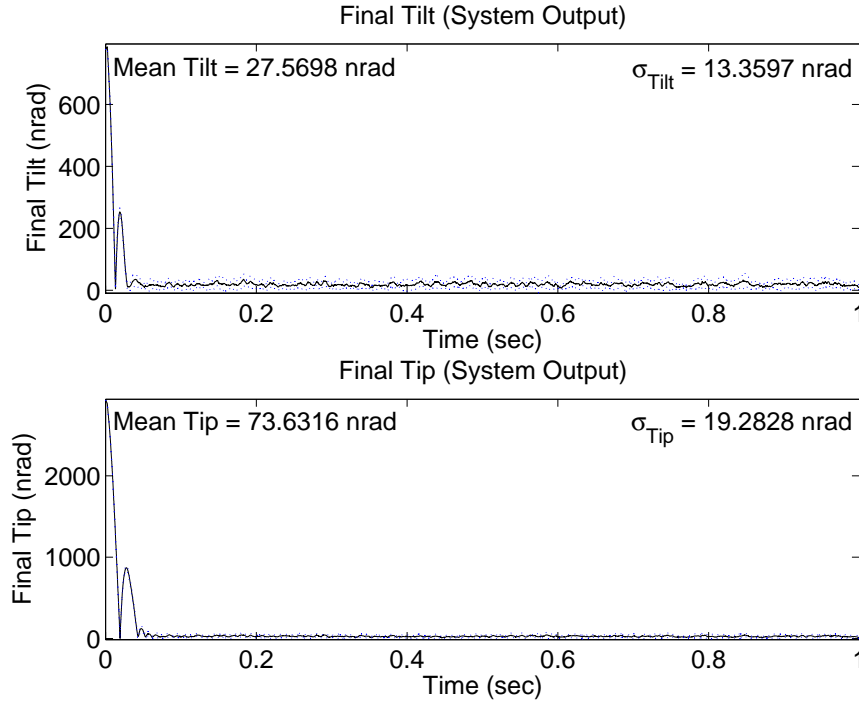


Figure 4.129 Case (20.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

noise only, as in Cases (15) and (16). Baseline Cases (13) and (14) showed that performance degradation did not occur when the LQG/LTR designs were simulated under nominal conditions. Similarly, Cases (21) and (22) show that there is no loss of performance using the LQG/LTR designs when sample time delays and sensor saturation are not present. Figures 4.130 to 4.135 show the mean and standard deviation values for Cases (21) and (22). Comparing the results to Cases (15) and (16) demonstrates that there is no degradation in performance when using the LQG/LTR designs. The mean and standard deviation values decrease in each case. As in Cases (19) and (20), Cases (21a) and (22a) improve the most, but subcase (c) exhibits the best performance.

4.2.2.5 Cases (23), (24), (25). The purpose of Cases (23), (24), and (25) is to evaluate the performance and robustness of the control system when operating for 4 seconds and subjected to tilt/tip disturbances defined by each of the three different Greenwood frequencies of 43 Hz, 86 Hz, and 189.2 Hz. Only the non-acquisition simulation runs are executed since the small input performance of the system is the primary focus

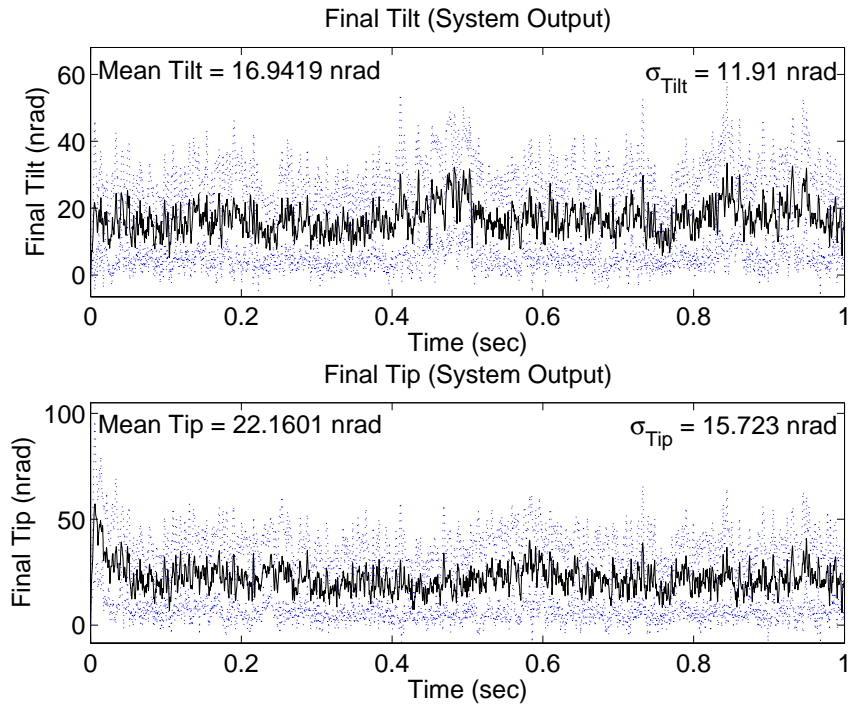


Figure 4.130 Case (21a): $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

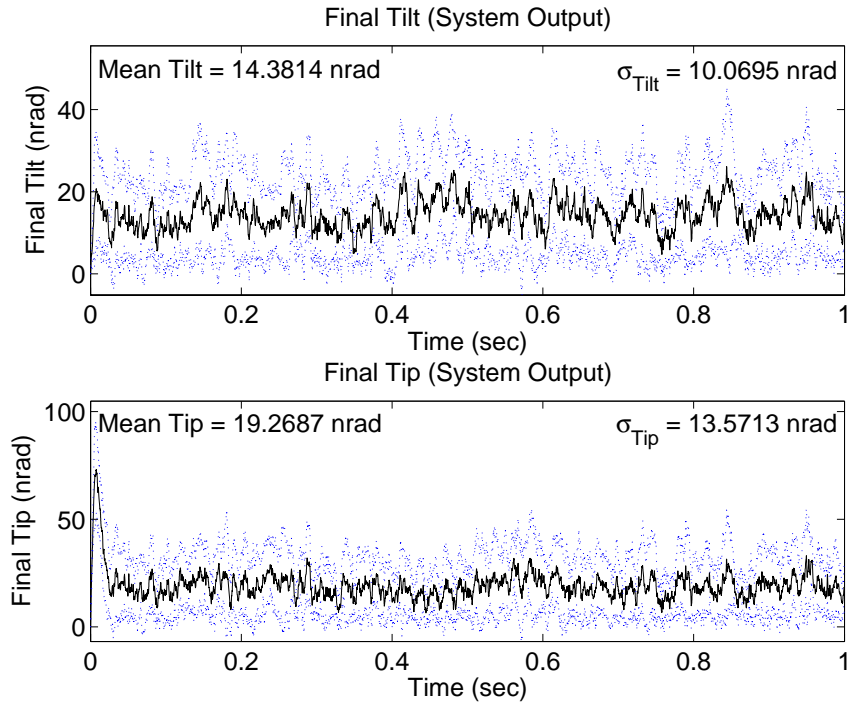


Figure 4.131 Case (21b): $\mathbf{U}_{diag} = [5E3 \quad 5E3 \quad 1E4 \quad 1E4]$

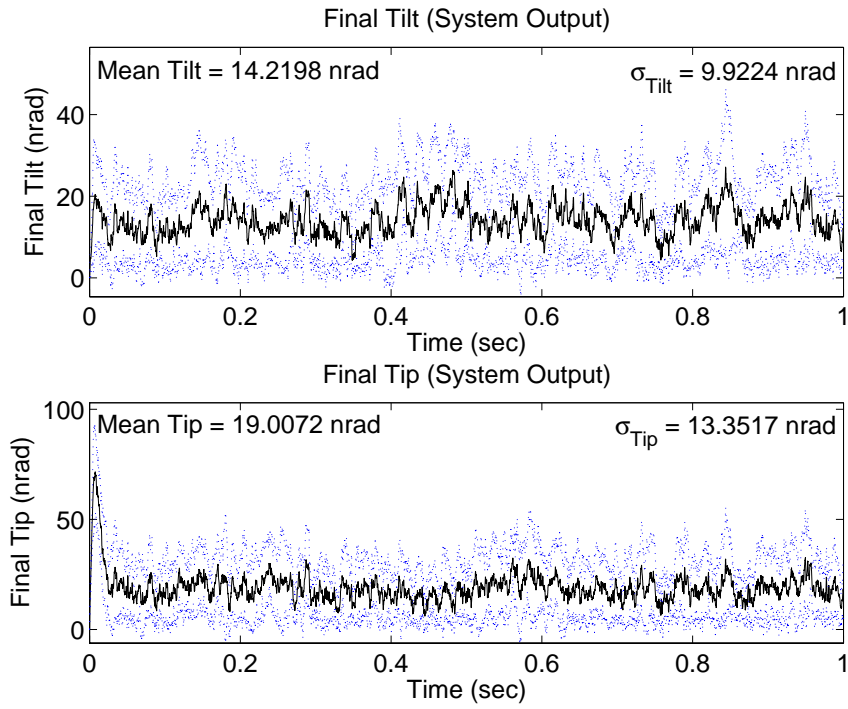


Figure 4.132 Case (21c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

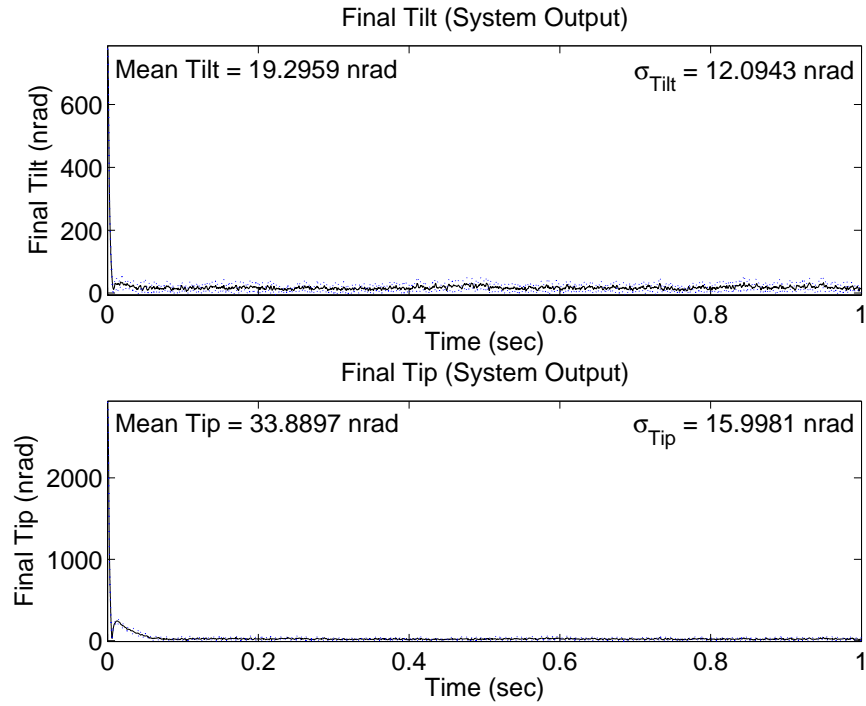


Figure 4.133 Case (22a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

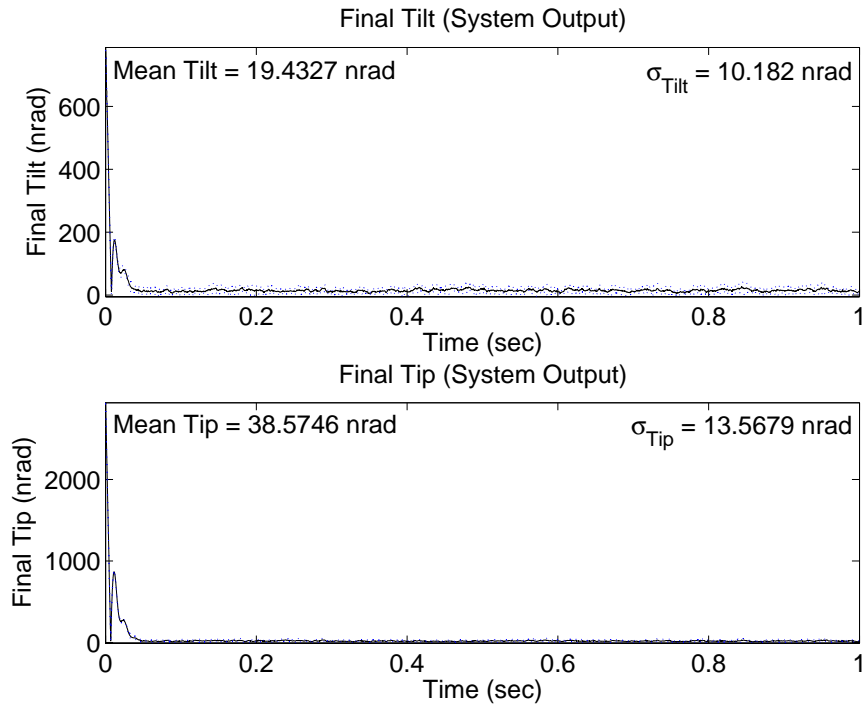


Figure 4.134 Case (22b): $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

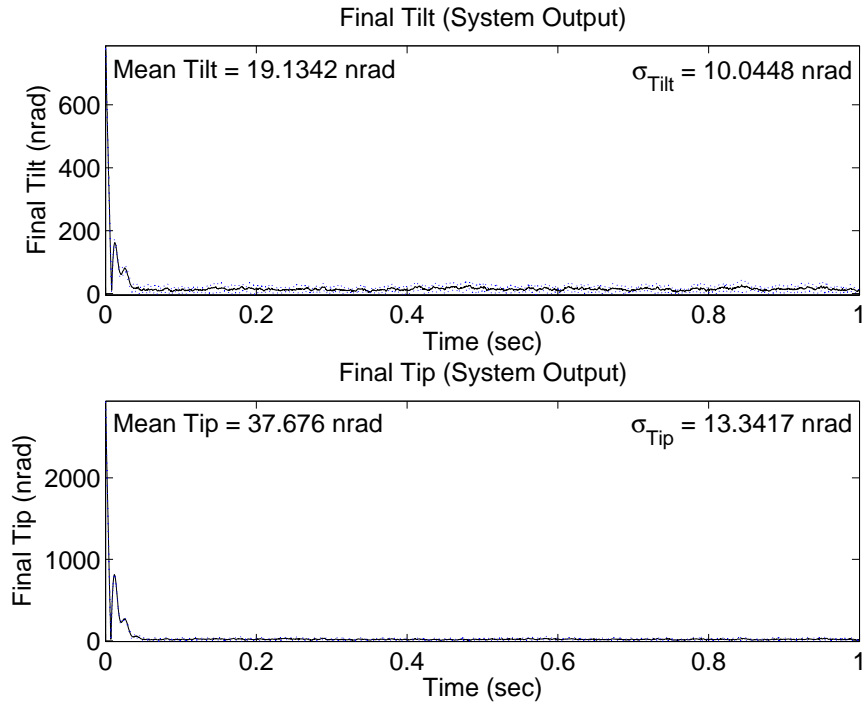


Figure 4.135 Case (22c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

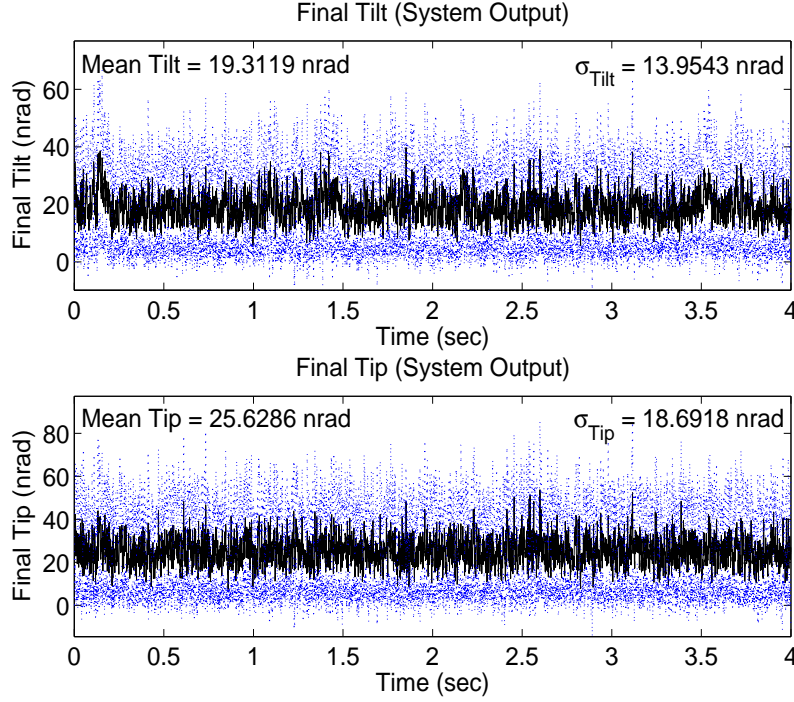


Figure 4.136 Case (23.5a): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

of this research. The three controller designs based on the LQG/LTR construct are used since the prior cases have demonstrated that these designs provide the best performance and robustness. Cases (23), (24), and (25) simulate the control loop with a five-sample time delay, as well as sensor saturation and noise as defined earlier in this chapter.

Case (23). Case (23.5) uses tilt and tip disturbance values based on a Greenwood frequency of 86 Hz; therefore, the case is identical to Case (17.5) except that the simulation is performed over 4 seconds. The performance and control inputs of Case (23.5) are shown in Figures 4.136 to 4.141. The control systems of Cases (23.5b) and (23.5c) outperform the system of Case (23.5a) in mean and standard deviation values. The performance in Case (23.5c) is slightly better than in Case (23.5b), but no more than 0.7 nrad separate the mean and standard deviation values in the two cases. As seen previously, the final tilt/tip values and control inputs of Case (23.5a) are much noisier than in Cases (23.5b) and (23.5c).

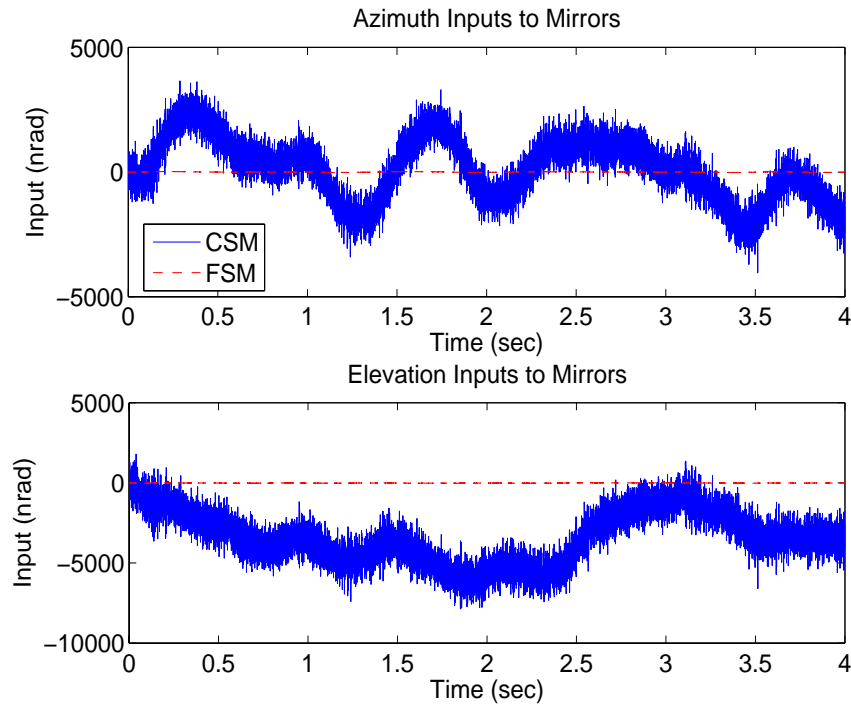


Figure 4.137 Case (23.5a): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 1 & 1 & 1E4 & 1E4 \end{bmatrix}$

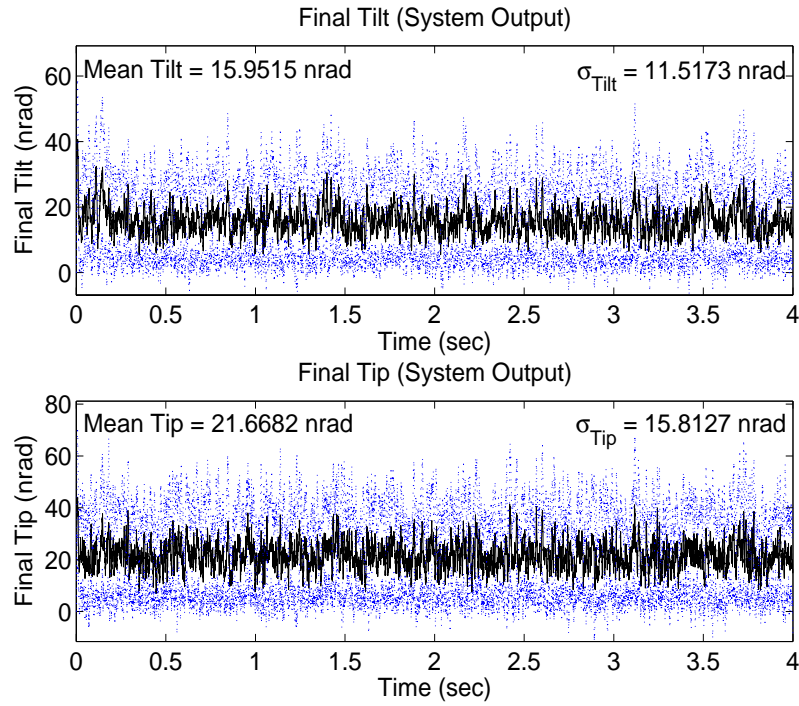


Figure 4.138 Case (23.5b): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = \begin{bmatrix} 5E3 & 5E3 & 1E4 & 1E4 \end{bmatrix}$

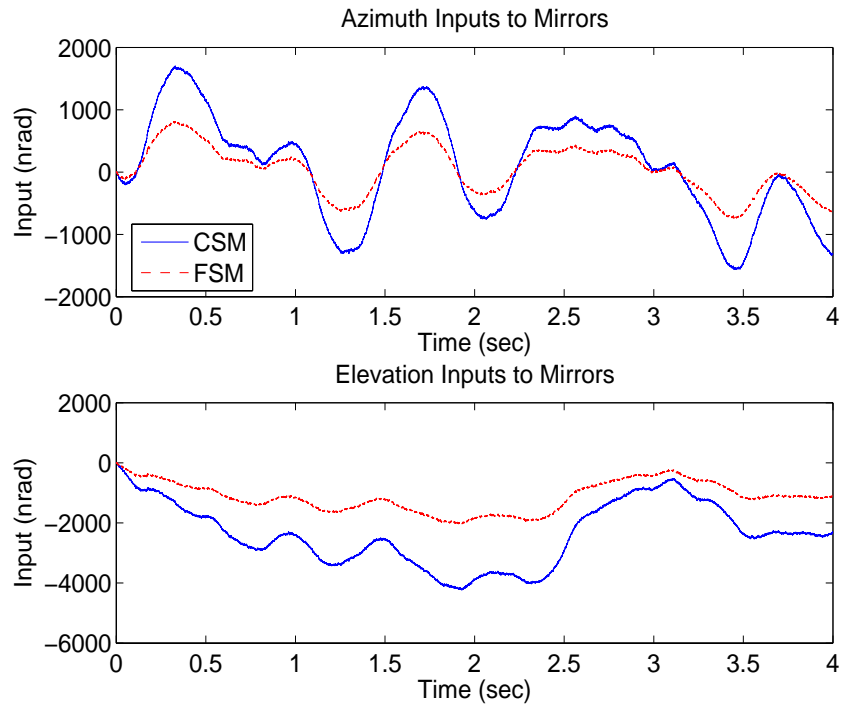


Figure 4.139 Case (23.5b): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

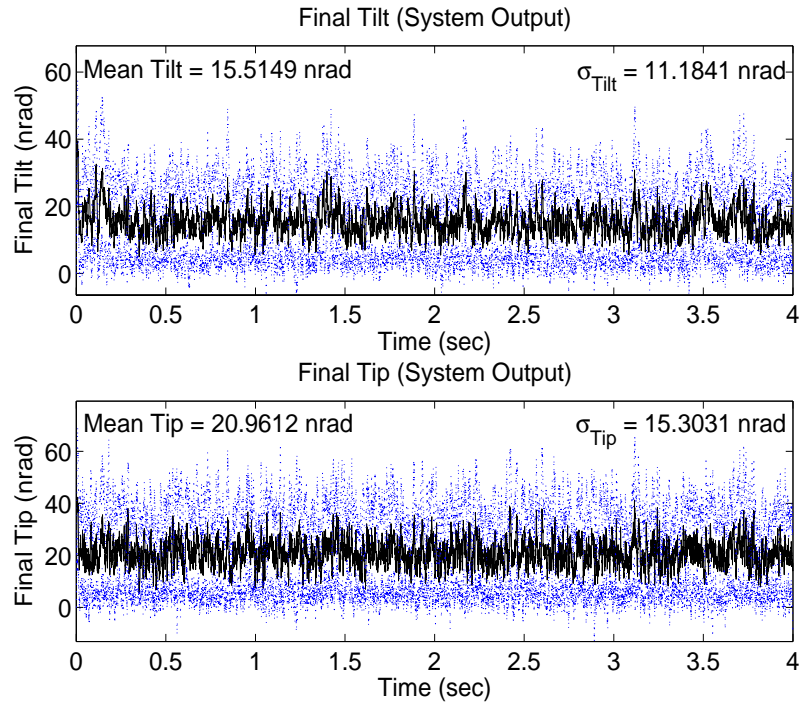


Figure 4.140 Case (23.5c): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

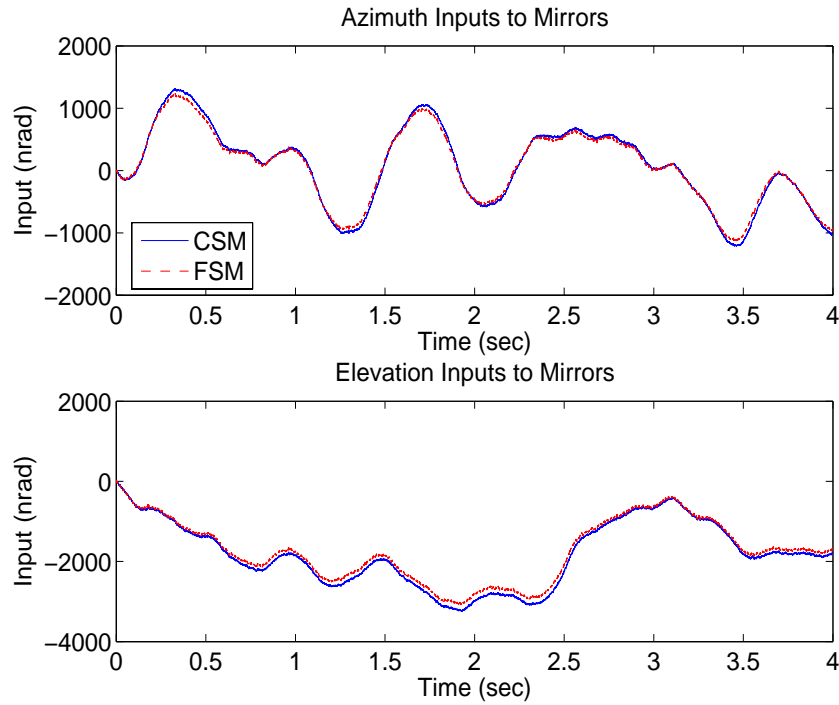


Figure 4.141 Case (23.5c): Delay 5, $f_G = 86 \text{ Hz}$, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

Case (24.5) uses tilt and tip disturbance values based on a Greenwood frequency of 189.2 Hz. Figures 4.142 to 4.147 display the performance and control inputs of Case (24.5). The same trends observed in Case (23.5) are evident in Case (24.5). Case (24.5a) shows the worst performance and Case (24.5c) is slightly better than Case (24.5b) (by no more than 0.8 nrad). Overall performance in Case (24.5) actually improves over Case (23.5), though not by much, even though the Greenwood frequency increases. In Cases (24.5b) and (24.5c), the mean and standard deviation values decrease by no more than 0.3 nrad compared to Case (23.5). The performance improvement may be due to the disturbance model correlation times which are based on an atmospheric wind speed of 21 m/s; the Greenwood frequency of 189.2 Hz is based on a wind speed of 22 m/s, while 86 Hz is based on 10 m/s.

Case (25.5) uses tilt and tip disturbance values based on a Greenwood frequency of 43 Hz. Figures 4.148 to 4.153 display the performance and control inputs of Case (25.5). Again, Case (25.5a) shows the worst performance and Case (25.5c) results in mean and

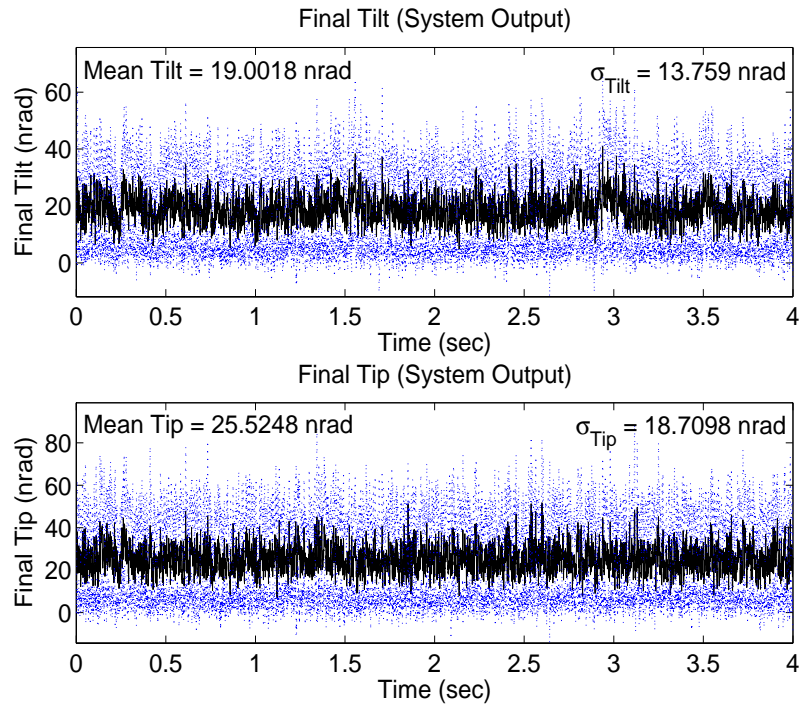


Figure 4.142 Case (24.5a): Delay 5, $f_G = 189.2$ Hz, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

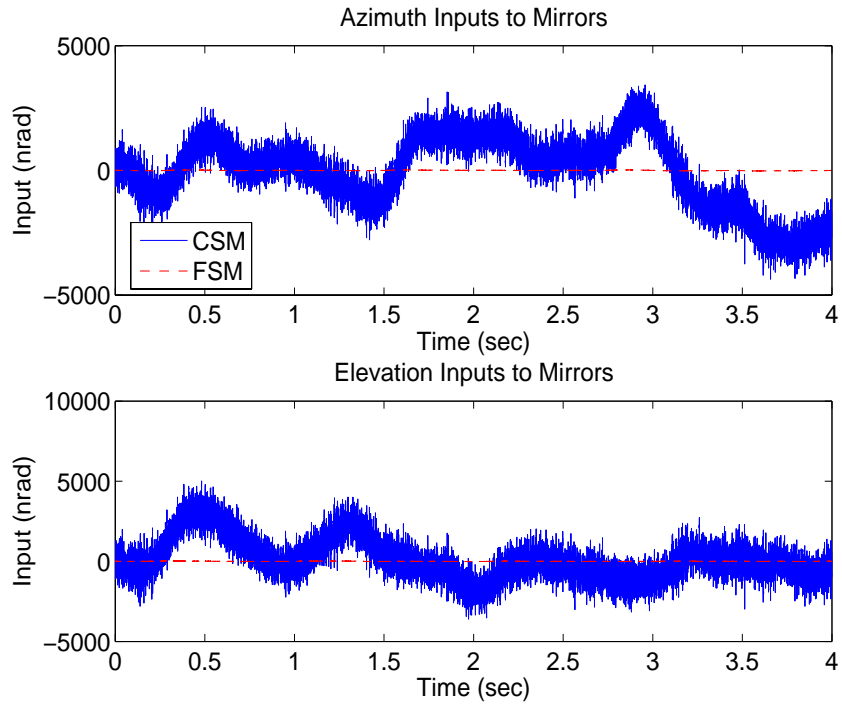


Figure 4.143 Case (24.5a): Delay 5, $f_G = 189.2$ Hz, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

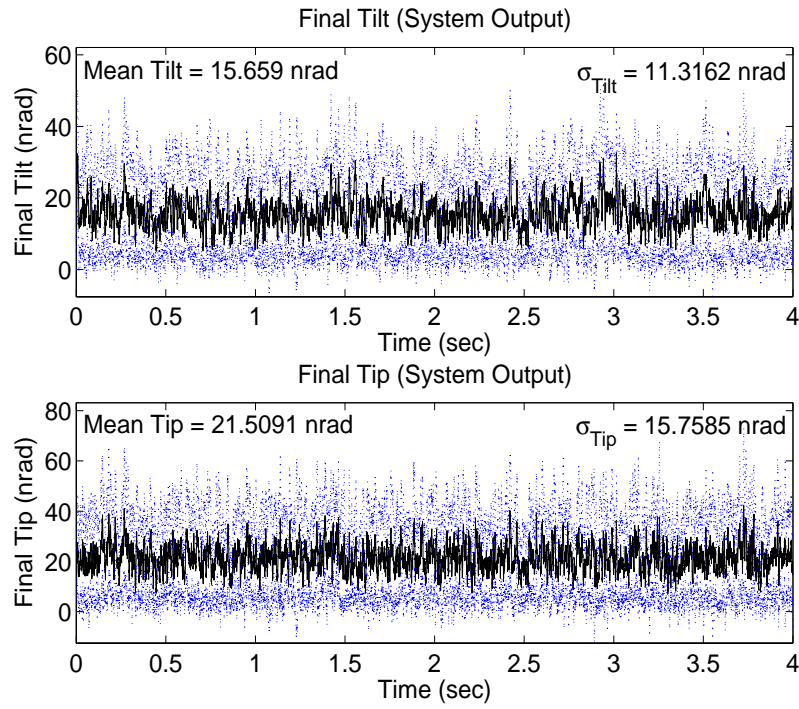


Figure 4.144 Case (24.5b): Delay 5, $f_G = 189.2 \text{ Hz}$, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

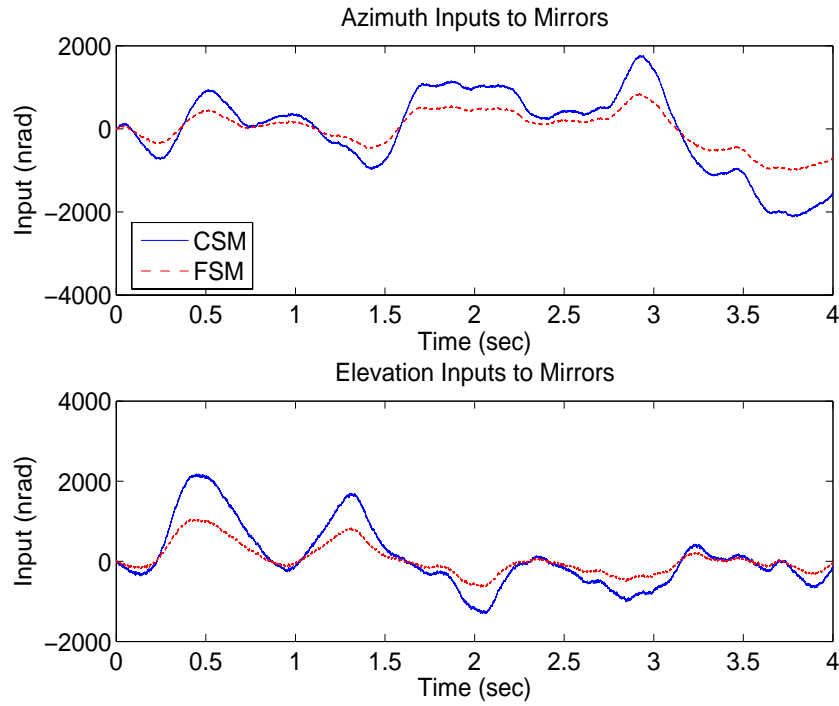


Figure 4.145 Case (24.5b): Delay 5, $f_G = 189.2 \text{ Hz}$, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

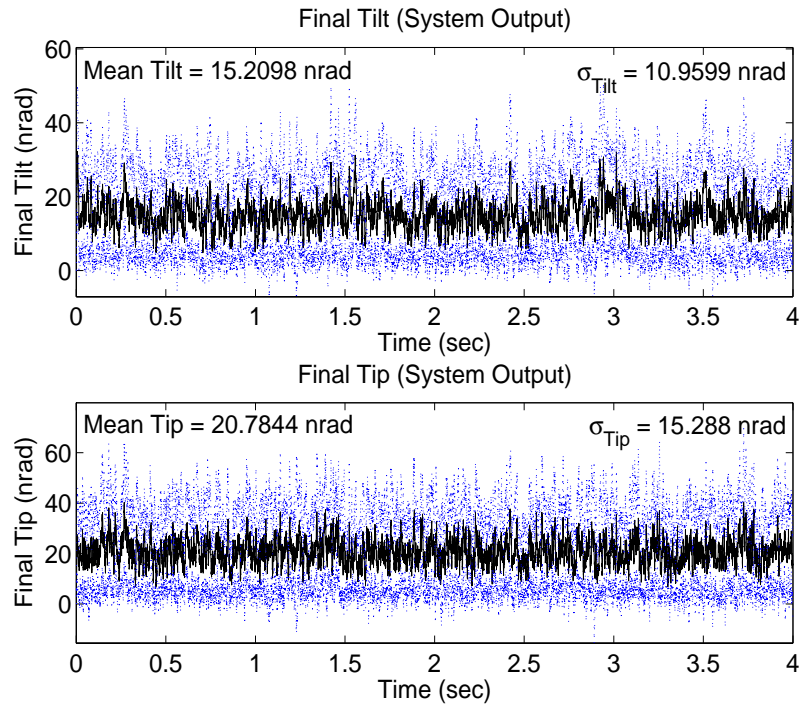


Figure 4.146 Case (24.5c): Delay 5, $f_G = 189.2 \text{ Hz}$, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

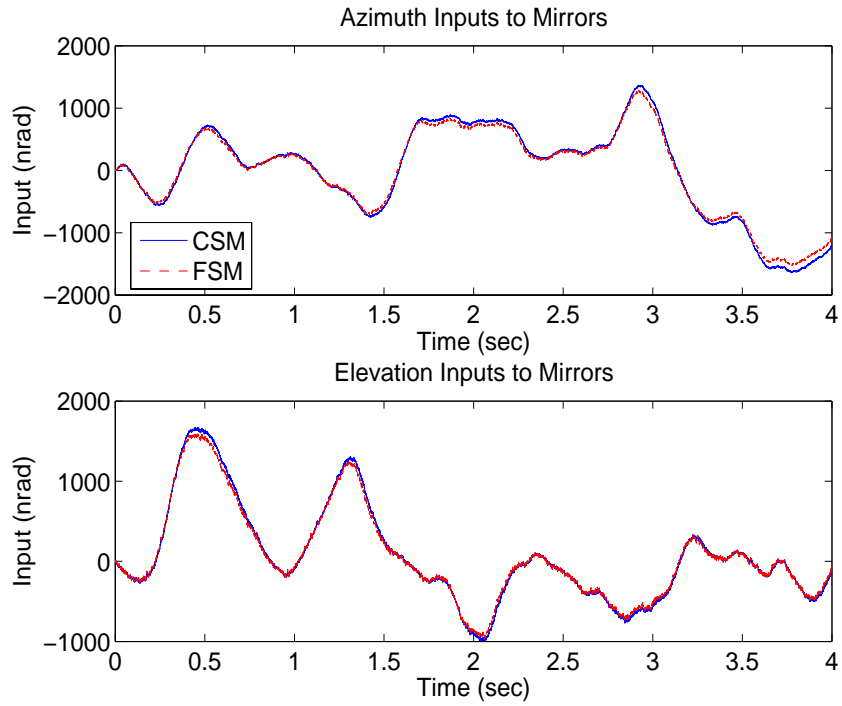


Figure 4.147 Case (24.5c): Delay 5, $f_G = 189.2 \text{ Hz}$, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

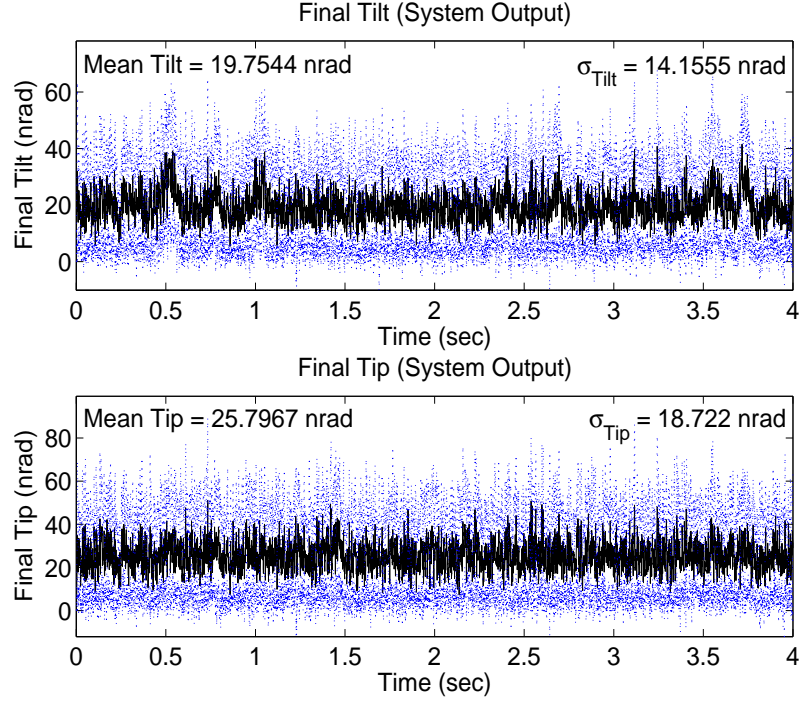


Figure 4.148 Case (25.5a): Delay 5, $f_G = 43$ Hz, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

standard deviation values that are no more than 0.7 nrad less than the values from Case (25.5b). The overall performance from Case (25.5) is worse than from Case (23.5), despite the lower Greenwood frequency, though the difference in azimuth mean and standard deviation values is no more than 0.4 nrad and elevation mean and standard deviation values is no more than 0.2 nrad. This may be due to the reason stated in Case (24.5), since 43 Hz is based on a wind speed of 5 m/s.

4.2.2.6 Case (26). The purpose of Case (26) is to repeat Case (15) using the unbiased variance values shown in Section 4.1.2.2, rather than the biased values used in all other cases, to determine if the performance is affected. Recall from Section 4.1.2.2 that the biased variance values are less than the unbiased values. Figures 4.154 to 4.156 show that the performance is minimally degraded. The mean values increase no more than 0.27 nrad and the standard deviation values increase by no more than 0.2 nrad. Therefore, the use of biased sensor noise variance values in simulation produces results accurate to within 0.27 nrad which is not a significant error.

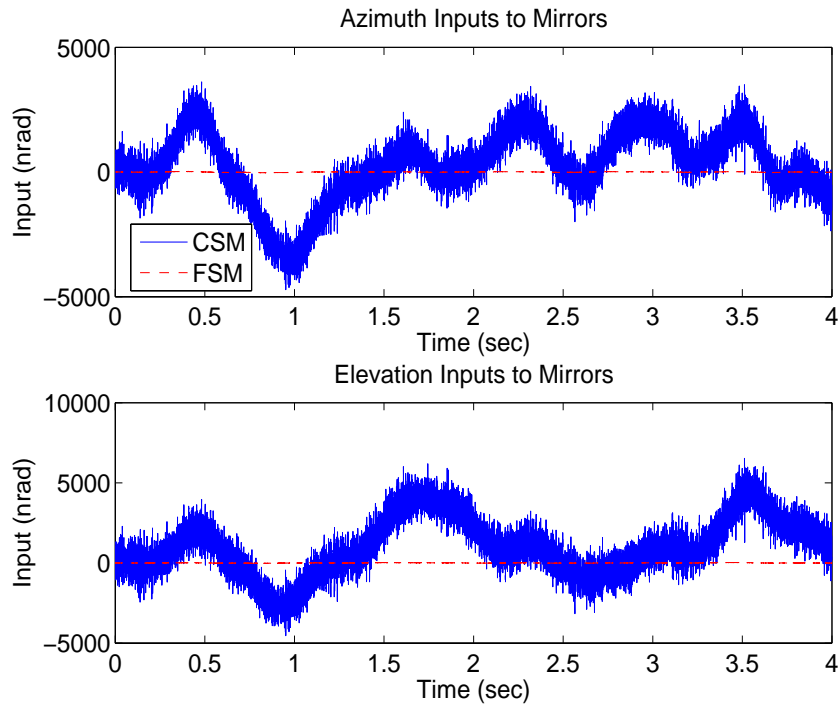


Figure 4.149 Case (25.5a): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

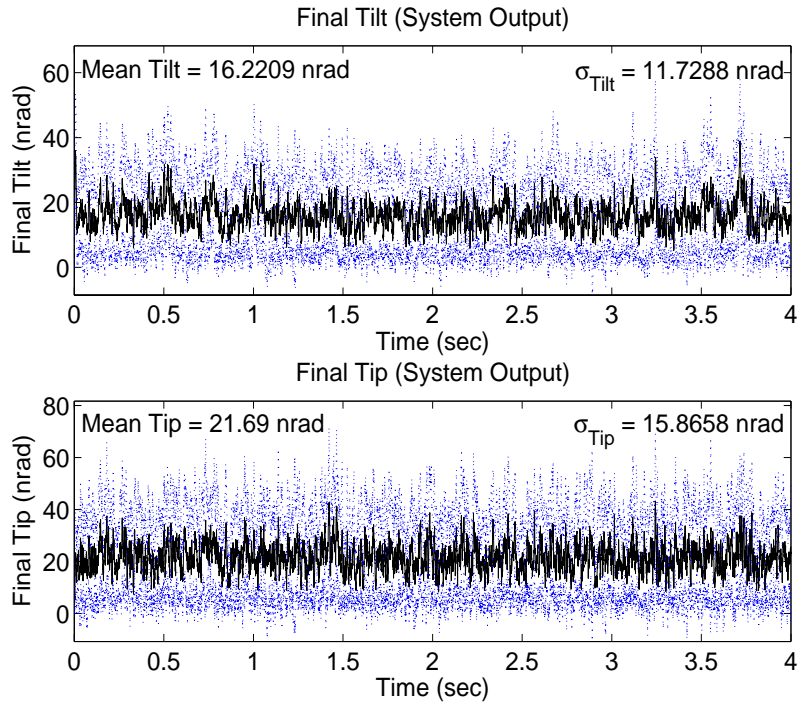


Figure 4.150 Case (25.5b): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

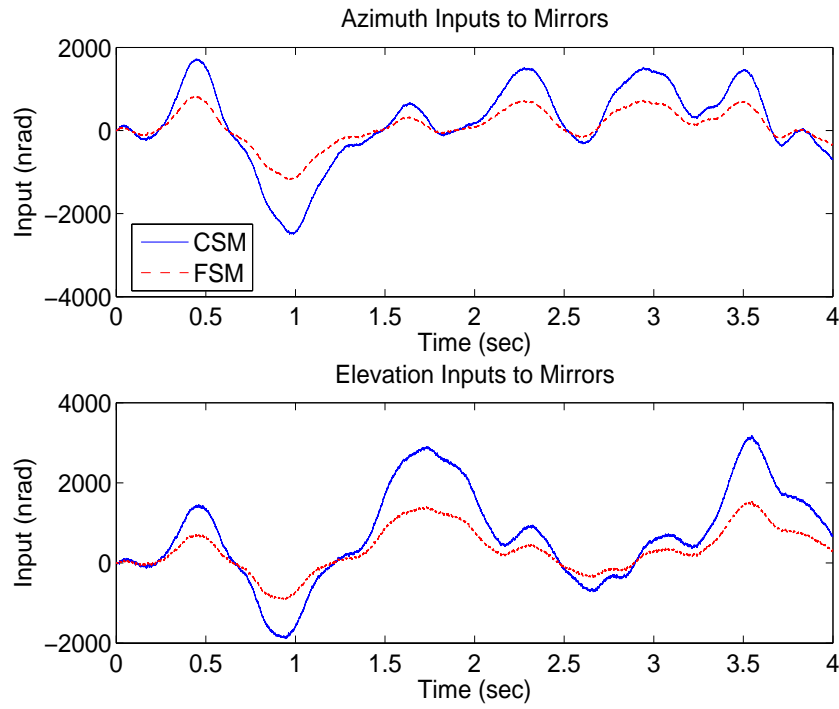


Figure 4.151 Case (25.5b): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

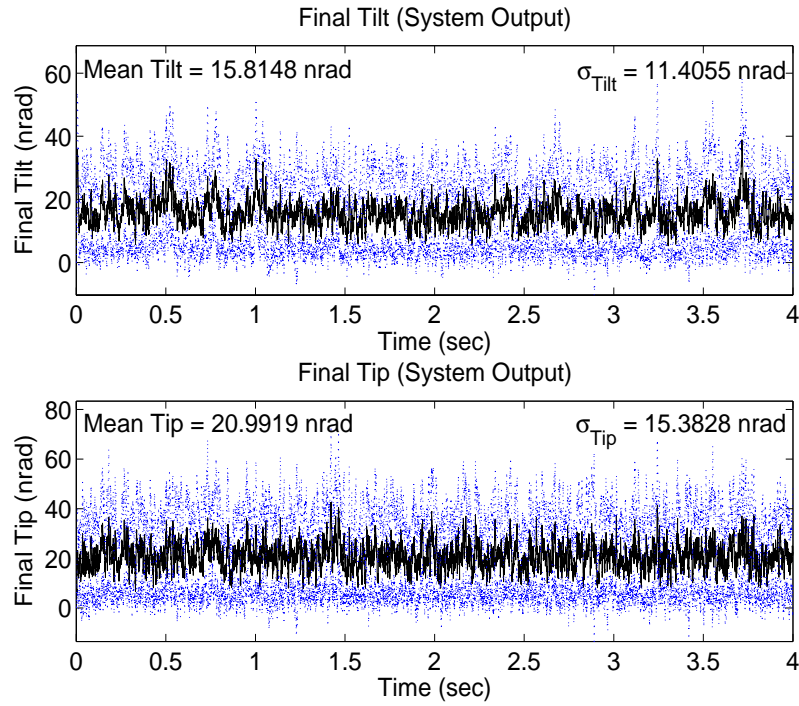


Figure 4.152 Case (25.5c): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

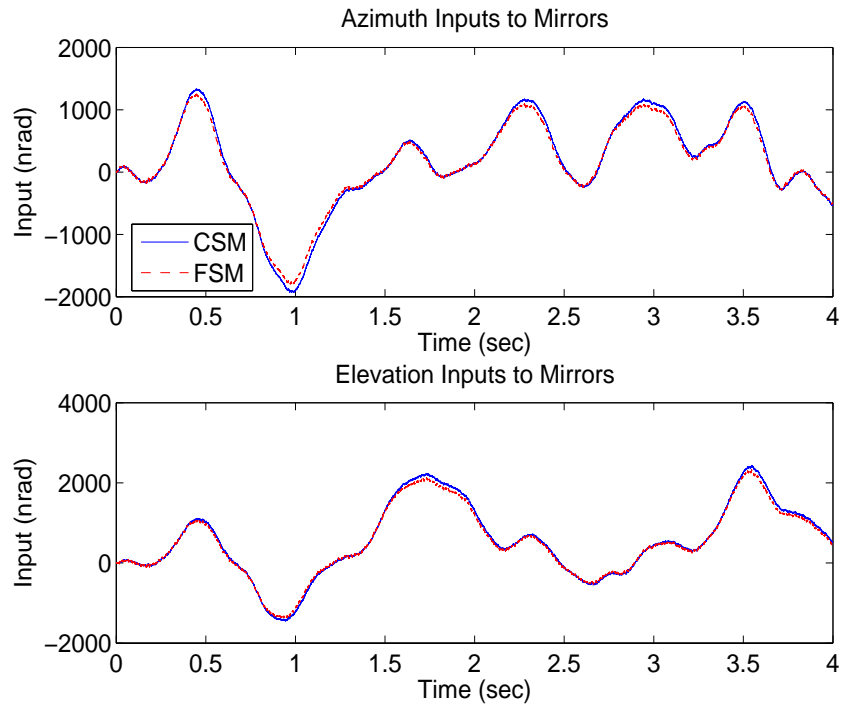


Figure 4.153 Case (25.5c): Delay 5, $f_G = 43 \text{ Hz}$, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

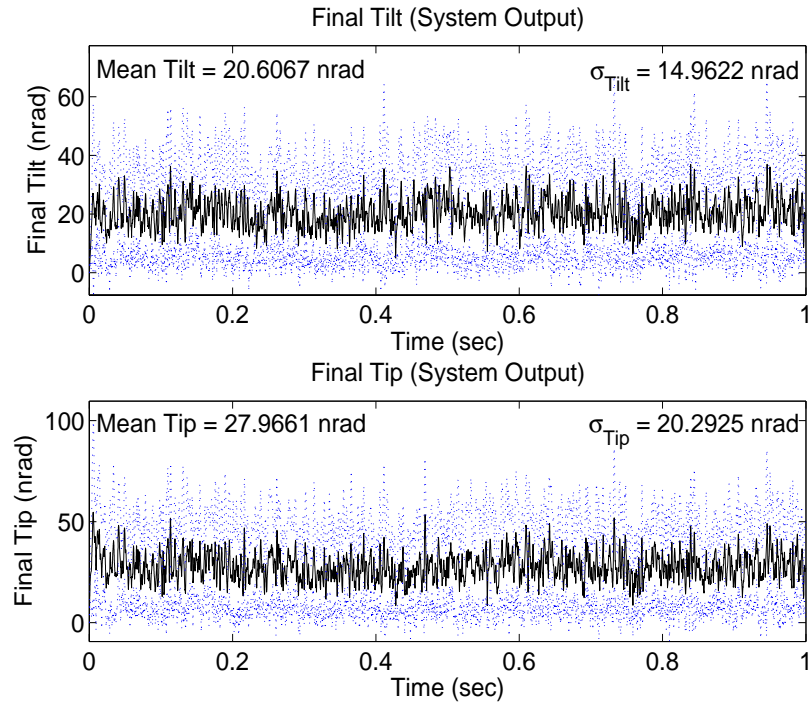


Figure 4.154 Case (26a): $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

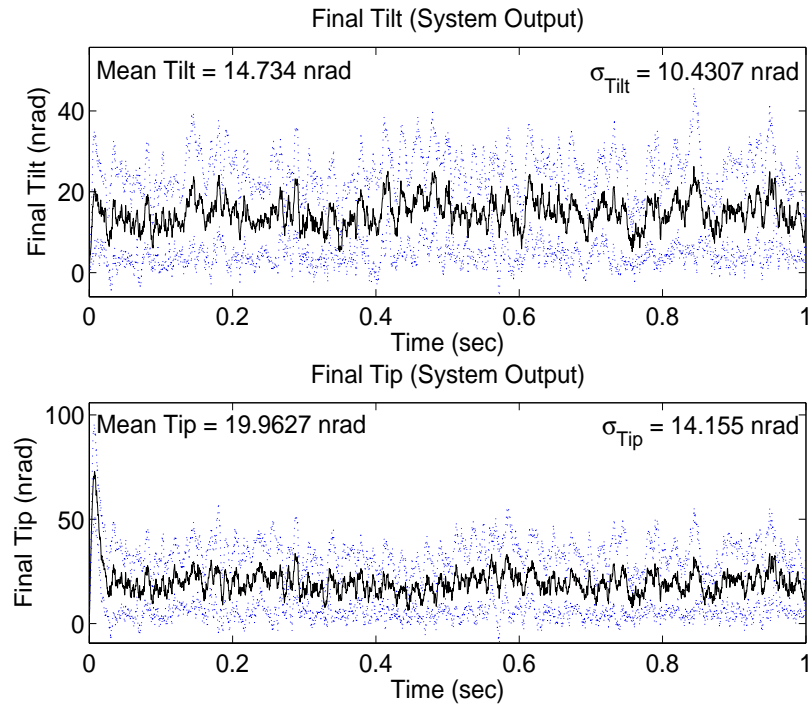


Figure 4.155 Case (26b): $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

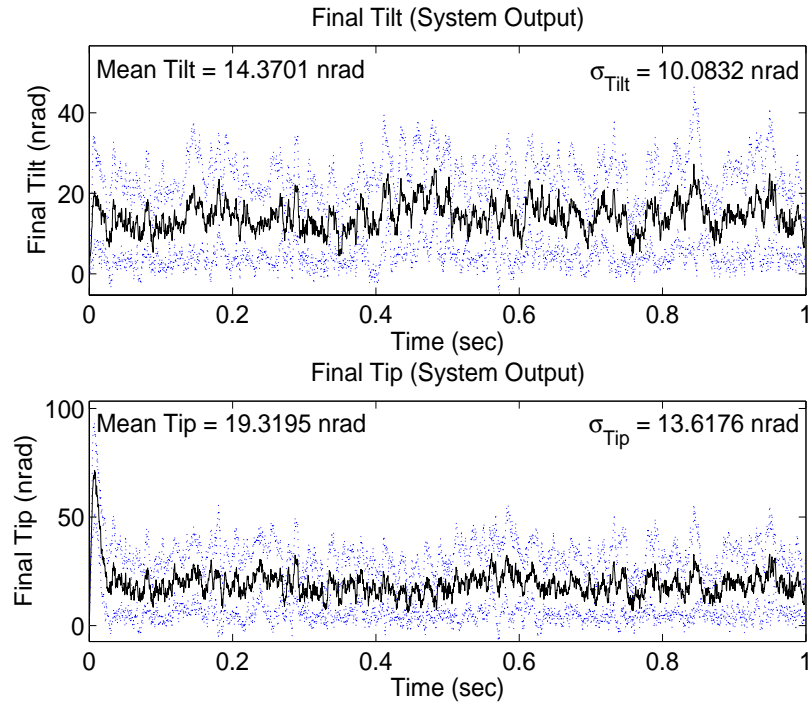


Figure 4.156 Case (26c): $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

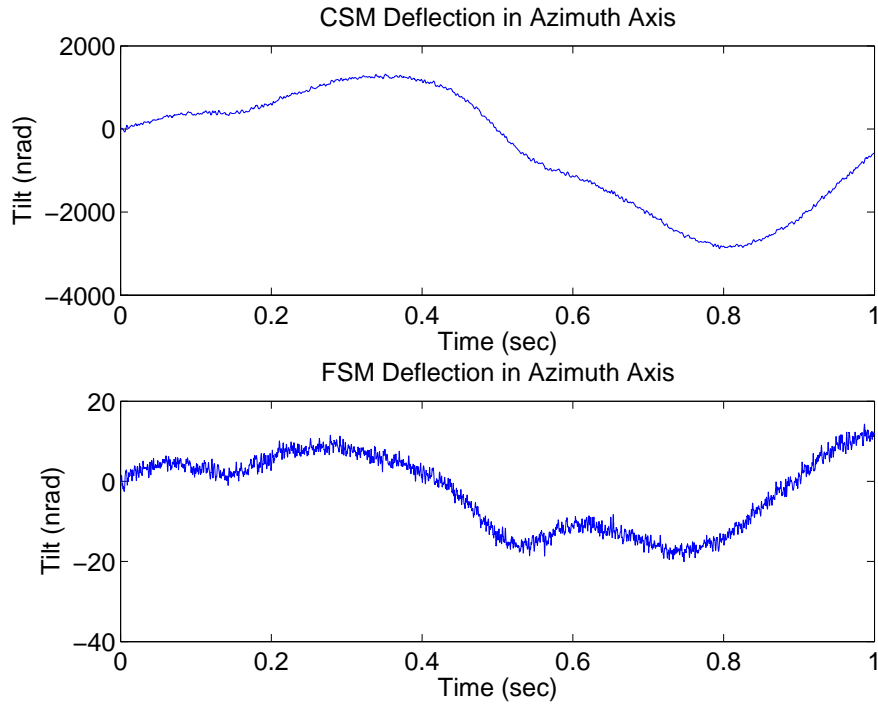


Figure 4.157 Case (15a): Steering Mirror Outputs (Azimuth Axis)

4.2.2.7 Steering Mirror Outputs. Figures 4.157 to 4.159 show the outputs of the CSM and FSM for Case (15). Note that the FSM tilt scaling in Figure 4.157 is much smaller than in the other two figures since most of the control authority in Case (1a) is given to the CSM. The control authority given by each design is evident from the decrease in CSM tilt and increase in FSM tilt from subcase (a) to subcase (c). Recall that the CSM is supposed to compensate for low-frequency disturbances, while the FSM compensates for high-frequency disturbances. These behaviors can be seen in each case in which the output of the FSM is changing faster than the CSM. Most importantly, the CSM output in subcase (a) is changing faster (noisier) than subcases (b) and (c) since the FSM does not have enough control authority to do its job completely. The poor performance of subcase (a) is probably due to the CSM attempting to compensate for higher frequency disturbances that the FSM is unable to bear due to its restricted control authority. The elevation outputs have the same characteristics and the plots are representative of all sensor-noise cases.

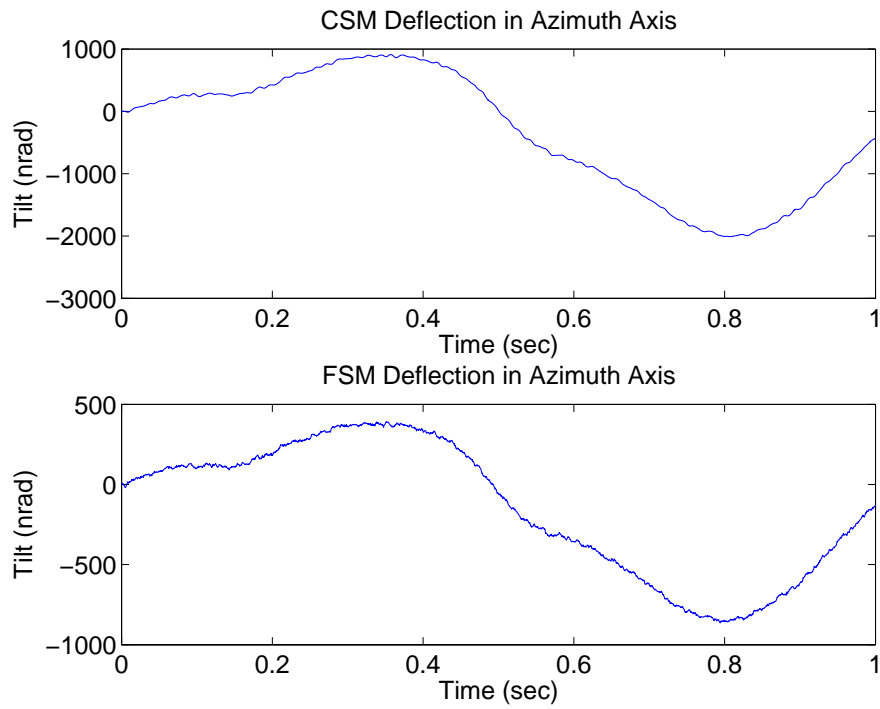


Figure 4.158 Case (15b): Steering Mirror Outputs (Azimuth Axis)

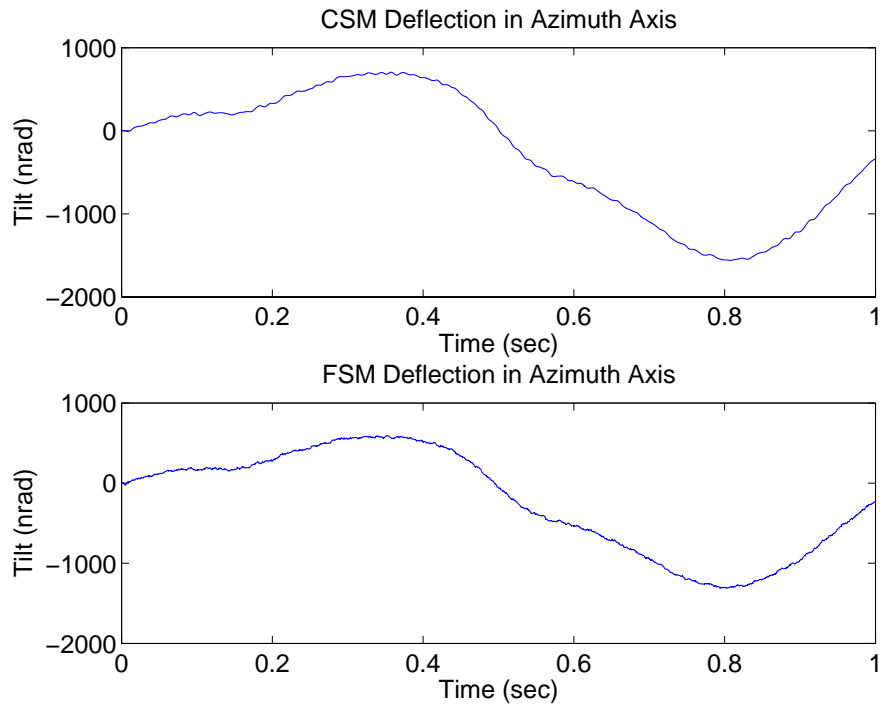


Figure 4.159 Case (15c): Steering Mirror Outputs (Azimuth Axis)

4.2.2.8 Summary of Sensor Noise Cases. With the presence of measurement sensor noise, saturation, and various time delays, the LQG designs of subcases (b) and (c) provide much better performance and robustness than the design of subcase (a). As in the baseline cases, time delays degrade the control systems' response times and performance; however, each subcase control system remains stable up to (and possibly beyond) a ten-sample time delay, potentially due to new tuning parameter values. The LTR technique improved the robustness of each subcase with no performance degradation. Case (26) shows that using the unbiased sensor noise variance in simulations, rather than the biased variance, does not have a significant impact on the results. Table 4.9 displays the final results from Cases (23) to (25) using the LQG/LTR designs, showing that subcase (c) provides best performance and robustness.

Table 4.9 LQG/LTR Performance

$f_G = 43 \text{ Hz}$				
Subcase	Mean Tilt (nrad)	σ_{Tilt} (nrad)	Mean Tip (nrad)	σ_{Tip} (nrad)
(a)	19.75	14.16	25.80	18.72
(b)	16.22	11.73	21.69	15.87
(c)	15.81	11.41	20.99	15.38
$f_G = 86 \text{ Hz}$				
Subcase	Mean Tilt (nrad)	σ_{Tilt} (nrad)	Mean Tip (nrad)	σ_{Tip} (nrad)
(a)	19.31	13.95	25.63	18.69
(b)	15.95	11.52	21.67	15.81
(c)	15.51	11.18	20.96	15.30
$f_G = 189.2 \text{ Hz}$				
Subcase	Mean Tilt (nrad)	σ_{Tilt} (nrad)	Mean Tip (nrad)	σ_{Tip} (nrad)
(a)	19.00	13.76	25.52	18.71
(b)	15.66	11.32	21.51	15.76
(c)	15.21	10.96	20.78	15.29

4.2.3 Additional Baseline Case. The following case determines if the added stability observed in Cases (17) and (18) is due to the modified \mathbf{Q} and \mathbf{R} matrices.

4.2.3.1 Case (27). The purpose of Case (27) is to evaluate the performance and robustness of the LQG control system under baseline conditions of Case (1) and (7) while using the \mathbf{Q} matrix from the sensor noise cases and the \mathbf{R} matrix from the baseline cases. Case (27.4a) simulates subcase (a) with a four-sample time delay to determine if using the modified \mathbf{Q} matrix achieves stable performance unlike Case (7.4a). Figure 4.160 shows the final tilt and tip values, and Figure 4.161 displays the optimal control inputs. It is evident from Figure 4.160 that the control system remains stable and the mean and standard deviation values are reasonable. The control inputs are identical to those of Case (1a). Case (27a) simulates subcase (a) with no delay, to compare directly with Case (1a). Figure 4.162 shows the final tilt and tip values in Case (27a). The performance is worse than in Case (1a) in terms of the mean and standard deviation values, but it does improve slightly over Case (27.4a) as expected since the delay is not present. The most significant change from Case (1a) is that the system is much more "sluggish", as shown by the much longer initial transient before the tilt/tip is brought to zero for the first time. The slow behavior of the system is present beyond the initial transient for the entire length of the simulation. Therefore, the use of the modified \mathbf{Q} matrix makes the system much more robust to time delays at the cost of decreased performance. Figure 4.163 shows the same control inputs as seen in Case (1a).

Figures 4.164 to 4.166 display the performance in Case (27.10) with a 10-sample time delay to show that the control system in each subcase remains stable as seen in the sensor noise cases, despite unstable performance shown in baseline Case (7.10). Cases (27.10b) and (27.10c) perform best in the elevation axis while Case (27.10a) performs better in the azimuth axis; however, Cases (27.10b) and (27.10c) are much more robust than Case (27.10a) with much less oscillation present.

4.3 Summary

This chapter presented the simulation, analysis, and results of three LQG/LTR designs that differ in the amount of control authority given to the CSM and FSM. The simulations are used to develop realistic atmospheric tilt and tip disturbances applied to a representative control loop to determine the performance and robustness of the controller

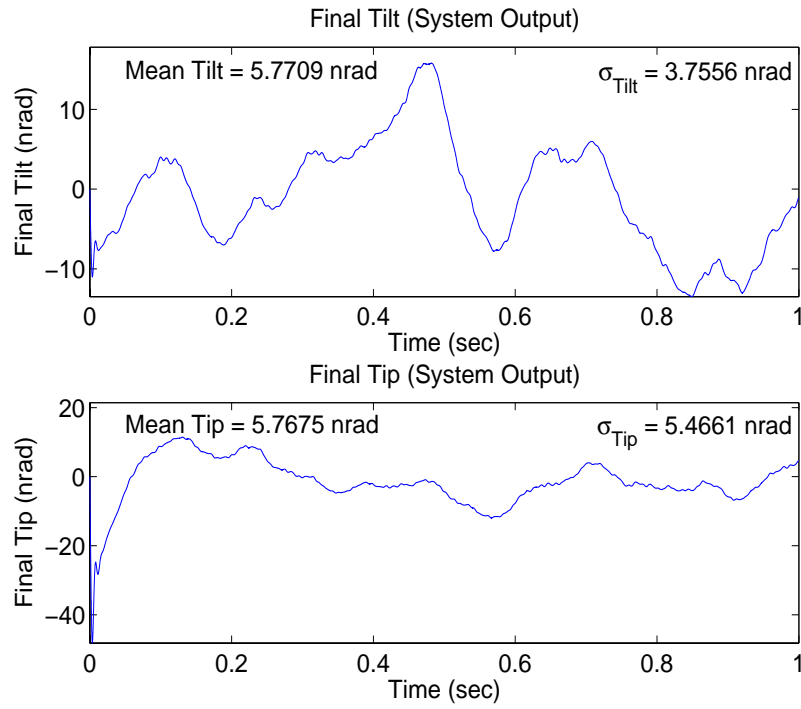


Figure 4.160 Case (27.4a): Delay 4, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

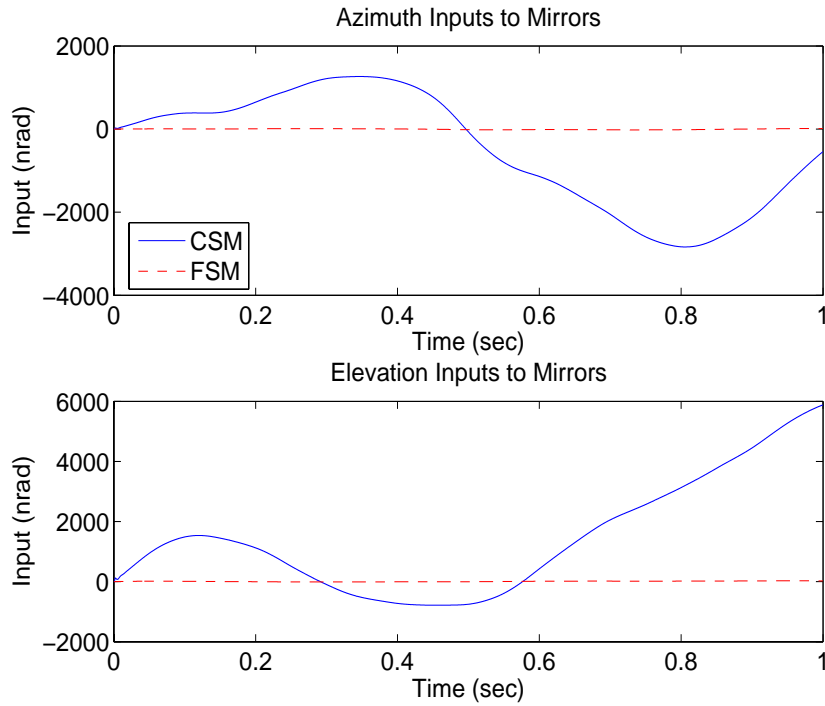


Figure 4.161 Case (27.4a): Delay 4, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

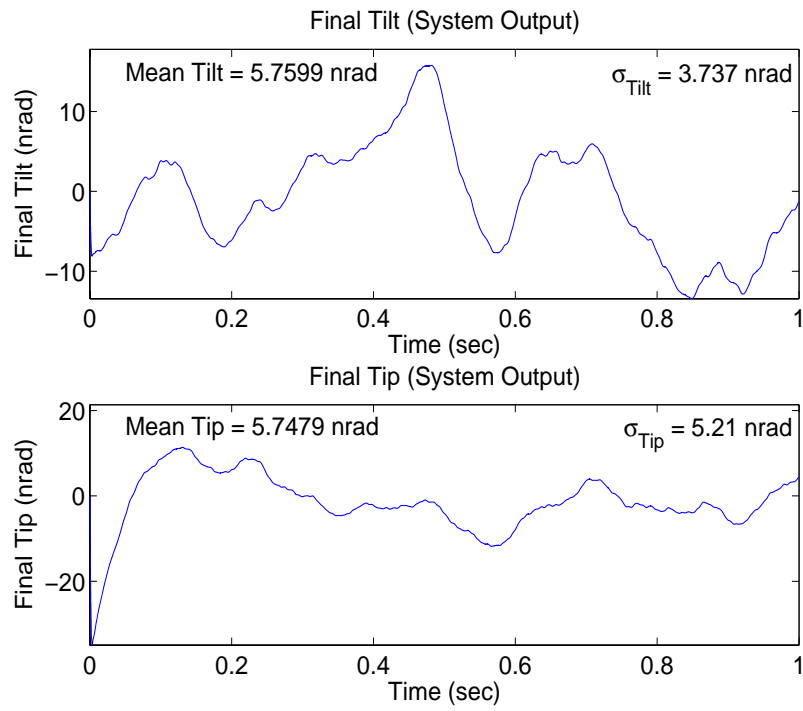


Figure 4.162 Case (27a): $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

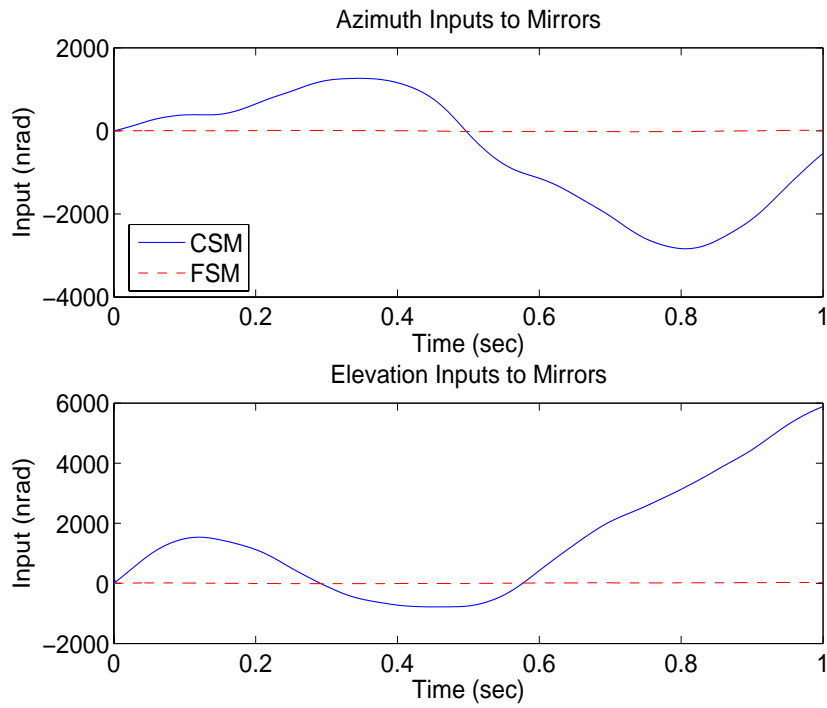


Figure 4.163 Case (27a): $\mathbf{U}_{diag} = [1 \quad 1 \quad 1E4 \quad 1E4]$

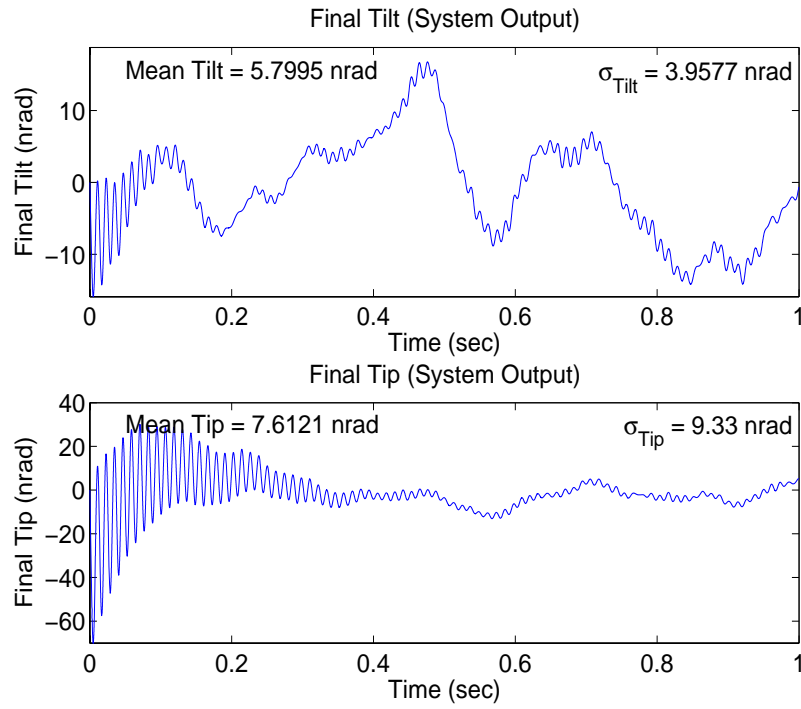


Figure 4.164 Case (27.10a): Delay 10, $\mathbf{U}_{diag} = [1 \ 1 \ 1E4 \ 1E4]$

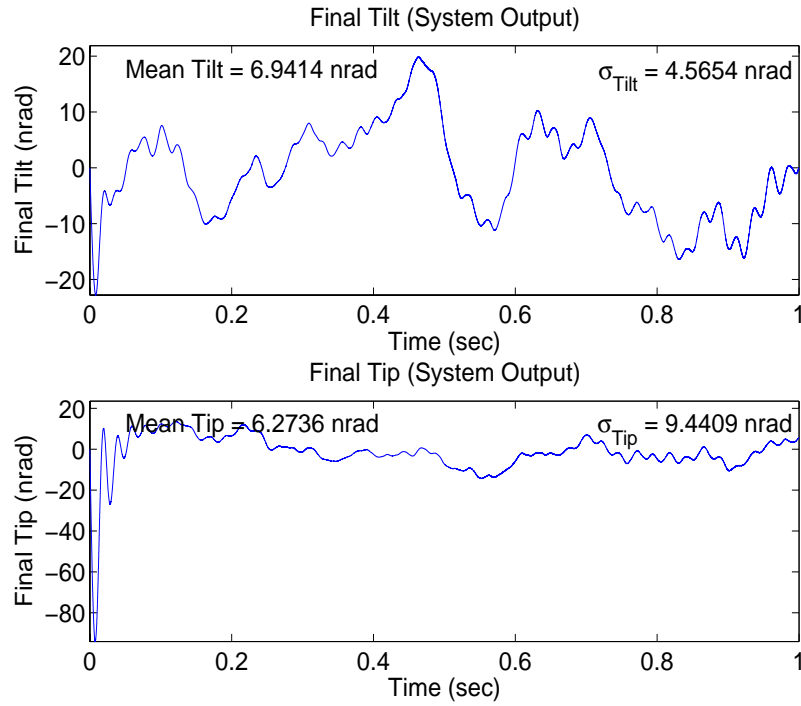


Figure 4.165 Case (27.10b): Delay 10, $\mathbf{U}_{diag} = [5E3 \ 5E3 \ 1E4 \ 1E4]$

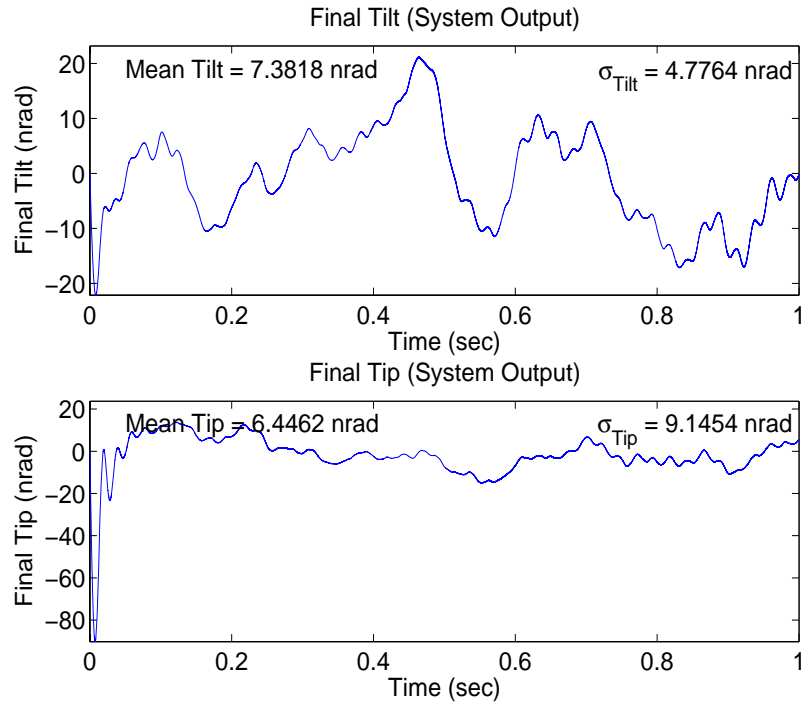


Figure 4.166 Case (27.10c): Delay 10, $\mathbf{U}_{diag} = [1E4 \ 1E4 \ 1E4 \ 1E4]$

designs. The control loop simulation accounts for sensor noise, sensor saturation, and system processing time. The next chapter discusses the conclusions drawn from the results and provides recommendations.

V. Conclusions and Recommendations

5.1 Summary

This research proposes the use of an LQG/LTR controller within the tracking loop of the SOR 3.5-m telescope to control the CSM and FSM simultaneously. The controller is designed such that the control system eliminates wavefront tilt and tip induced by the turbulence of earth's atmosphere. This is accomplished by estimating the atmospheric tilt and tip disturbance and controlling the angular orientation of the steering mirrors to match the negative of the estimates. The LQG controller consists of a Kalman filter and LQR-based tracker designed separately via the separation principle. Linear models of the CSM and FSM were developed from experimental frequency response data. The Kalman filter incorporates of the CSM and FSM models, a tilt and tip disturbance model, and pyramid sensor measurement model to develop state estimates provided to the LQR. The tilt and tip disturbance model is a first-order lag driven by WGN based on the variance and correlation time of the atmospheric tilt and tip. The pyramid sensor provides noisy, saturated measurements to the Kalman filter. The LQR is based upon of the same CSM, FSM, and tilt disturbance models as well as two additional pseudointegral states (to provide type-1 versus type-0 control, essential to accomplishing the performance goals). The pseudointegral states represent the integral of the regulation error, defined as the summation of the steering mirrors' output and wavefront tilt and tip on the incoming light. By regulating the pseudointegral states to zero, the output of the steering mirrors track the negative of the wavefront tilt and tip to correct the phase distortion. An LTR technique is applied to the LQG design to increase the controller's robustness to the sensor noise and system time delays.

The performance and robustness of the LQG and LQG/LTR designs is determined through simulation of the tracking control loop. A separate "truth model" simulation develops angular atmospheric tilt and tip values based on the von Karman spectrum and the Fourier series phase screen method. The angular tilt and tip values are used in the control loop simulation to serve as the actual wavefront tilt and tip. Performance and robustness of the controller designs is evaluated from results of numerous simulation cases

with varying parameters such as atmospheric turbulence strength, system time delays, sensor noise, sensor saturation, and run time. Three LQG designs that differ only in the amount of control authority given to each steering mirror are evaluated in each simulation case. The three designs are referred to as subcases (a), (b), and (c). The design of subcase (a) allocates nearly all of the control authority to the CSM, while subcase (c) provides nearly equal authority to the CSM and FSM. The design of subcase (b) provides roughly four times more control authority to the CSM than the FSM. The LTR technique is applied to each of the designs and evaluated in simulations.

5.2 Conclusions

This section presents the significant conclusions drawn from the simulation results and analysis of Chapter 4.

- The LQG/LTR design of subcase (c), with nearly equal control authority given to the CSM and FSM, provides the best performance and robustness in the presence of pyramid sensor noise, sensor saturation, and time delays (the statistics of subcase (c) are italicized in the following tables for this reason). The LQG/LTR design of subcase (b) provides performance and robustness very similar to subcase (c), while the LQG/LTR design of subcase (a) provides degraded performance and robustness. The performance statistics of each LQG/LTR design from a one-second simulation with sensor noise only and Greenwood frequency, $f_G = 86 \text{ Hz}$ (average turbulence conditions), are

Sensor Noise Only				
	$f_G = 86 \text{ Hz}$			
Subcase	Mean Tilt (nrad)	σ_{Tilt} (nrad)	Mean Tip (nrad)	σ_{Tip} (nrad)
(a)	16.94	11.91	22.16	15.72
(b)	14.38	10.07	19.27	13.57
(c)	<i>14.22</i>	<i>9.92</i>	<i>19.01</i>	<i>13.35</i>

The performance statistics of each LQG/LTR design from a one-second simulation with sensor noise, sensor saturation, and a ten-sample time delay for $f_G = 86 \text{ Hz}$ are

Sensor Noise & Sat. & Ten-Sample Delay				
	$f_G = 86 \text{ Hz}$			
Subcase	Mean Tilt (nrad)	σ_{Tilt} (nrad)	Mean Tip (nrad)	σ_{Tip} (nrad)
(a)	24.38	17.77	33.97	24.68
(b)	19.63	14.03	28.66	20.39
(c)	<i>18.79</i>	<i>13.40</i>	<i>27.24</i>	<i>19.33</i>

The performance statistics of each LQG/LTR design from a four-second simulation with sensor noise, sensor saturation, and a five-sample time delay from a four-second simulation for each Greenwood frequency are

Sensor Noise & Sat. & Five-Sample Delay				
	$f_G = 43 \text{ Hz}$			
Subcase	Mean Tilt (nrad)	σ_{Tilt} (nrad)	Mean Tip (nrad)	σ_{Tip} (nrad)
(a)	19.75	14.16	25.80	18.72
(b)	16.22	11.73	21.69	15.87
(c)	<i>15.81</i>	<i>11.41</i>	<i>20.99</i>	<i>15.38</i>

Sensor Noise & Sat. & Five-Sample Delay				
	$f_G = 86 \text{ Hz}$			
Subcase	Mean Tilt (nrad)	σ_{Tilt} (nrad)	Mean Tip (nrad)	σ_{Tip} (nrad)
(a)	19.31	13.95	25.63	18.69
(b)	15.95	11.52	21.67	15.81
(c)	<i>15.51</i>	<i>11.18</i>	<i>20.96</i>	<i>15.30</i>

Sensor Noise & Sat. & Five-Sample Delay				
	$f_G = 189.2 \text{ Hz}$			
Subcase	Mean Tilt (nrad)	σ_{Tilt} (nrad)	Mean Tip (nrad)	σ_{Tip} (nrad)
(a)	19.00	13.76	25.52	18.71
(b)	15.66	11.32	21.51	15.76
(c)	<i>15.21</i>	<i>10.96</i>	<i>20.78</i>	<i>15.29</i>

- The poor performance of subcase (a), when sensor noise is present, is probably due to the CSM attempting to compensate for higher frequency disturbances that the FSM is unable to bear due to its restricted control authority.
- The LTR technique of modifying the state weighting matrix provides improved performance in the presence of sensor noise and makes the LQG design more robust to system time delays. However, this suggests that the original, unmodified cost function may not be tuned properly (the state weighting matrix in particular) for desired performance, since performance improvement due to LTR is not expected.
- Simulations showed that the LQG/LTR designs are able to regulate tilt and tip disturbances at least as large as 3000 nrad, even in the presence of sensor noise, sensor saturation, and system time delays.
- The baseline simulations showed that system time delays slow down the response of the controller designs and degrade performance. However, the sensor saturation actually improves the slow response by removing high frequency oscillations.
- The performance of the controller designs is verified to be almost identical in the azimuth and elevation axes. The results show worse performance in the elevation axis due to higher sensor noise and different tilt and tip disturbance values.
- Increasing the \mathbf{Q} matrix (dynamics noise strength) diagonal elements slows the response of the control system and degrades performance; however, robustness to system time delays is improved drastically. This is similar to the LTR technique of adding process noise to the Kalman filter model to recover robustness from the LQR.

- The best overall performance was attained when $f_G = 189.2 \text{ Hz}$ (high turbulence conditions), with an atmospheric average wind speed of 22 m/s, since the correlation times within the tilt/tip disturbance model are based on a wind speed of 21 m/s.

5.3 Recommendations

This section presents recommendations to expand upon the current research and analysis discussed.

- The system model and truth model used in the controller design and simulation are reduced-order models, although an eighteen-order model was developed that matched the CSM and FSM dynamics data very well. A study is strongly recommended to determine the effects of using the reduced-order model in the controller design versus the eighteen-order truth model, and the effects of using the eighteen-order model for both.
- Anti-windup compensation should be incorporated in the Proportional-plus-Integral tracker design.
- Further frequency domain analysis of the control system could be performed to determine the error rejection capabilities.
- Rate limiters should be incorporated into the simulation truth model to represent dynamics limitations of the CSM and FSM actuators.
- Further performance analysis of the Kalman filter could be conducted and used to benefit proper tuning choices.
- A study could be performed to examine the effects of lower sampling rates which would result in larger differences in disturbance values, from sample time to sample time, to correct for.
- Analysis of using time-varying control and Kalman filter gains could be performed to compare to the results from this research which used constant steady-state control and Kalman filter gains. Using time-varying gains should improve performance, but the study would reveal the level of improvement possible.

- Appropriate tilt and tip correlation times for the various Greenwood frequencies could be used in the disturbance model, rather than using one value for all frequencies. A look-up table or parameter estimation could be used in the implementation.
- Additional sources of tilt and tip could be modeled and simulated rather than only atmosphere-induced disturbances. Other sources could include off-center viewing of the imaged object, mount jitter, and imperfect optics.
- As mentioned earlier, the dynamics of the steering mirrors change depending on the size of the input. Therefore, gain scheduling could be incorporated based on knowledge of the inputs supplied by the controller. An additional option is to develop a nonlinear model of the steering mirror dynamics and design a nonlinear controller.

Appendix A. System Model Matrices

A.1 Continuous-Time Plant State-Space Matrices

A.1.1 Modified Jordan Canonical Form

. Matrix **A**:

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8
row 1	-1006.69	0	0	0	0	0	0	0
row 2	0	-65.18	380.05	0	0	0	0	0
row 3	0	-380.05	-65.18	0	0	0	0	0
row 4	0	0	0	-1006.69	0	0	0	0
row 5	0	0	0	0	-65.18	380.05	0	0
row 6	0	0	0	0	-380.05	-65.18	0	0
row 7	0	0	0	0	0	0	-4441.15	33514.23
row 8	0	0	0	0	0	0	-33514.23	-4441.15
row 9	0	0	0	0	0	0	0	0
row 10	0	0	0	0	0	0	0	0
row 11	0	0	0	0	0	0	0	0
row 12	0	0	0	0	0	0	0	0
row 13	0	0	0	0	0	0	0	0
row 14	0	0	0	0	0	0	0	0
row 15	0	0	0	0	0	0	0	0
row 16	0	0	0	0	0	0	0	0
row 17	0	0	0	0	0	0	0	0
row 18	0	0	0	0	0	0	0	0
row 19	0	0	0	0	0	0	0	0
row 20	0	0	0	0	0	0	0	0
row 21	0	0	0	0	0	0	0	0
row 22	0	0	0	0	0	0	0	0
row 23	0	0	0	0	0	0	0	0
row 24	0	0	0	0	0	0	0	0
row 25	0	0	0	0	0	0	0	0
row 26	0	0	0	0	0	0	0	0
row 27	0	0	0	0	0	0	0	0

	col 9	col 10	col 11	col 12	col 13	col 14	col 15
row 1	0	0	0	0	0	0	0
row 2	0	0	0	0	0	0	0
row 3	0	0	0	0	0	0	0
row 4	0	0	0	0	0	0	0
row 5	0	0	0	0	0	0	0
row 6	0	0	0	0	0	0	0
row 7	0	0	0	0	0	0	0
row 8	0	0	0	0	0	0	0
row 9	-4327.99	22893.05	0	0	0	0	0
row 10	-22893.05	-4327.99	0	0	0	0	0
row 11	0	0	-14720.15	0	0	0	0
row 12	0	0	0	-97.50	8849.86	0	0
row 13	0	0	0	-8849.86	-97.50	0	0
row 14	0	0	0	0	0	-7021.19	0
row 15	0	0	0	0	0	0	-984.71
row 16	0	0	0	0	0	0	-4541.65
row 17	0	0	0	0	0	0	0
row 18	0	0	0	0	0	0	0
row 19	0	0	0	0	0	0	0
row 20	0	0	0	0	0	0	0
row 21	0	0	0	0	0	0	0
row 22	0	0	0	0	0	0	0
row 23	0	0	0	0	0	0	0
row 24	0	0	0	0	0	0	0
row 25	0	0	0	0	0	0	0
row 26	0	0	0	0	0	0	0
row 27	0	0	0	0	0	0	0

	col 16	col 17	col 18	col 19	col 20	col 21	col 22
row 1	0	0	0	0	0	0	0
row 2	0	0	0	0	0	0	0
row 3	0	0	0	0	0	0	0
row 4	0	0	0	0	0	0	0
row 5	0	0	0	0	0	0	0
row 6	0	0	0	0	0	0	0
row 7	0	0	0	0	0	0	0
row 8	0	0	0	0	0	0	0
row 9	0	0	0	0	0	0	0
row 10	0	0	0	0	0	0	0
row 11	0	0	0	0	0	0	0
row 12	0	0	0	0	0	0	0
row 13	0	0	0	0	0	0	0
row 14	0	0	0	0	0	0	0
row 15	4541.65	0	0	0	0	0	0
row 16	-984.71	0	0	0	0	0	0
row 17	0	-5408.15	38008.56	0	0	0	0
row 18	0	-38008.56	-5408.15	0	0	0	0
row 19	0	0	0	-8059.96	29995.10	0	0
row 20	0	0	0	-29995.10	-8059.96	0	0
row 21	0	0	0	0	0	-3471.96	20540.62
row 22	0	0	0	0	0	-20540.62	-3471.96
row 23	0	0	0	0	0	0	0
row 24	0	0	0	0	0	0	0
row 25	0	0	0	0	0	0	0
row 26	0	0	0	0	0	0	0
row 27	0	0	0	0	0	0	0

	col 23	col 24	col 25	col 26	col 27
row 1	0	0	0	0	0
row 2	0	0	0	0	0
row 3	0	0	0	0	0
row 4	0	0	0	0	0
row 5	0	0	0	0	0
row 6	0	0	0	0	0
row 7	0	0	0	0	0
row 8	0	0	0	0	0
row 9	0	0	0	0	0
row 10	0	0	0	0	0
row 11	0	0	0	0	0
row 12	0	0	0	0	0
row 13	0	0	0	0	0
row 14	0	0	0	0	0
row 15	0	0	0	0	0
row 16	0	0	0	0	0
row 17	0	0	0	0	0
row 18	0	0	0	0	0
row 19	0	0	0	0	0
row 20	0	0	0	0	0
row 21	0	0	0	0	0
row 22	0	0	0	0	0
row 23	-156.64	8659.32	0	0	0
row 24	-8659.32	-156.64	0	0	0
row 25	0	0	-5242.84	0	0
row 26	0	0	0	-775.34	4518.22
row 27	0	0	0	-4518.22	-775.34

Matrix **B**:

	col 1	col 2	col 3	col 4
row 1	-36.22	0	0	0
row 2	11.18	0	0	0
row 3	-27.71	0	0	0
row 4	0	-36.22	0	0
row 5	0	11.18	0	0
row 6	0	-27.71	0	0
row 7	0	0	754.42	0
row 8	0	0	-1351.54	0
row 9	0	0	-1050.74	0
row 10	0	0	-288.54	0
row 11	0	0	-179.56	0
row 12	0	0	-71.73	0
row 13	0	0	14.59	0
row 14	0	0	-51.71	0
row 15	0	0	-20.45	0
row 16	0	0	38.57	0
row 17	0	0	0	7975.53
row 18	0	0	0	5730.95
row 19	0	0	0	9389.04
row 20	0	0	0	-7936.25
row 21	0	0	0	-3140.05
row 22	0	0	0	1068.08
row 23	0	0	0	166.09
row 24	0	0	0	-74.47
row 25	0	0	0	-33.78
row 26	0	0	0	-60.95
row 27	0	0	0	-21.48

Matrix \mathbf{C}^T :

	col 1	col 2
row 1	-4.01	0
row 2	-12.98	0
row 3	0	0
row 4	0	-4.01
row 5	0	-12.98
row 6	0	0
row 7	0.78	0
row 8	-1.46	0
row 9	-1.34	0
row 10	0.99	0
row 11	108.52	0
row 12	0.64	0
row 13	-1.14	0
row 14	-335.82	0
row 15	24.08	0
row 16	-81.54	0
row 17	0	0.27
row 18	0	-0.46
row 19	0	0.12
row 20	0	0.69
row 21	0	-0.59
row 22	0	-0.22
row 23	0	-0.46
row 24	0	0.43
row 25	0	-194.38
row 26	0	47.36
row 27	0	15.32

A.2 Kalman Filter Model State-Space Matrices

A.2.1 Equivalent Discrete-Time

. Matrix Φ :

	col 1	col 2	col 3	col 4	col 5	col 6	col 7	col 8
row 1	0.81764	0	0	0	0	0	0	0
row 2	0	0.98420	0.07495	0	0	0	0	0
row 3	0	-0.07495	0.98420	0	0	0	0	0
row 4	0	0	0	0.81764	0	0	0	0
row 5	0	0	0	0	0.98420	0.07495	0	0
row 6	0	0	0	0	-0.07495	0.98420	0	0
row 7	0	0	0	0	0	0	0.37569	0.16762
row 8	0	0	0	0	0	0	-0.16762	0.37569
row 9	0	0	0	0	0	0	0	0
row 10	0	0	0	0	0	0	0	0
row 11	0	0	0	0	0	0	0	0
row 12	0	0	0	0	0	0	0	0
row 13	0	0	0	0	0	0	0	0
row 14	0	0	0	0	0	0	0	0
row 15	0	0	0	0	0	0	0	0
row 16	0	0	0	0	0	0	0	0
row 17	0	0	0	0	0	0	0	0
row 18	0	0	0	0	0	0	0	0
row 19	0	0	0	0	0	0	0	0
row 20	0	0	0	0	0	0	0	0
row 21	0	0	0	0	0	0	0	0
row 22	0	0	0	0	0	0	0	0
row 23	0	0	0	0	0	0	0	0
row 24	0	0	0	0	0	0	0	0
row 25	0	0	0	0	0	0	0	0
row 26	0	0	0	0	0	0	0	0
row 27	0	0	0	0	0	0	0	0
row 28	0	0	0	0	0	0	0	0
row 29	0	0	0	0	0	0	0	0

	col 9	col 10	col 11	col 12	col 13	col 14	col 15	col 16
row 1	0	0	0	0	0	0	0	0
row 2	0	0	0	0	0	0	0	0
row 3	0	0	0	0	0	0	0	0
row 4	0	0	0	0	0	0	0	0
row 5	0	0	0	0	0	0	0	0
row 6	0	0	0	0	0	0	0	0
row 7	0	0	0	0	0	0	0	0
row 8	0	0	0	0	0	0	0	0
row 9	-0.05613	-0.41704	0	0	0	0	0	0
row 10	0.41704	-0.05613	0	0	0	0	0	0
row 11	0	0	0.05265	0	0	0	0	0
row 12	0	0	0	-0.19404	0.96130	0	0	0
row 13	0	0	0	-0.96130	-0.19404	0	0	0
row 14	0	0	0	0	0	0.24555	0	0
row 15	0	0	0	0	0	0	0.50511	0.64753
row 16	0	0	0	0	0	0	-0.64753	0.50511
row 17	0	0	0	0	0	0	0	0
row 18	0	0	0	0	0	0	0	0
row 19	0	0	0	0	0	0	0	0
row 20	0	0	0	0	0	0	0	0
row 21	0	0	0	0	0	0	0	0
row 22	0	0	0	0	0	0	0	0
row 23	0	0	0	0	0	0	0	0
row 24	0	0	0	0	0	0	0	0
row 25	0	0	0	0	0	0	0	0
row 26	0	0	0	0	0	0	0	0
row 27	0	0	0	0	0	0	0	0
row 28	0	0	0	0	0	0	0	0
row 29	0	0	0	0	0	0	0	0

	col 17	col 18	col 19	col 20	col 21	col 22	col 23	col 24
row 1	0	0	0	0	0	0	0	0
row 2	0	0	0	0	0	0	0	0
row 3	0	0	0	0	0	0	0	0
row 4	0	0	0	0	0	0	0	0
row 5	0	0	0	0	0	0	0	0
row 6	0	0	0	0	0	0	0	0
row 7	0	0	0	0	0	0	0	0
row 8	0	0	0	0	0	0	0	0
row 9	0	0	0	0	0	0	0	0
row 10	0	0	0	0	0	0	0	0
row 11	0	0	0	0	0	0	0	0
row 12	0	0	0	0	0	0	0	0
row 13	0	0	0	0	0	0	0	0
row 14	0	0	0	0	0	0	0	0
row 15	0	0	0	0	0	0	0	0
row 16	0	0	0	0	0	0	0	0
row 17	0.08463	0.32831	0	0	0	0	0	0
row 18	-0.32831	0.08463	0	0	0	0	0	0
row 19	0	0	0.19149	-0.05593	0	0	0	0
row 20	0	0	0.05593	0.19149	0	0	0	0
row 21	0	0	0	0	-0.28373	-0.41095	0	0
row 22	0	0	0	0	0.41095	-0.28373	0	0
row 23	0	0	0	0	0	0	-0.15543	0.95661
row 24	0	0	0	0	0	0	-0.95661	-0.15543
row 25	0	0	0	0	0	0	0	0
row 26	0	0	0	0	0	0	0	0
row 27	0	0	0	0	0	0	0	0
row 28	0	0	0	0	0	0	0	0
row 29	0	0	0	0	0	0	0	0

	col 25	col 26	col 27	col 28	col 29
row 1	0	0	0	0	0
row 2	0	0	0	0	0
row 3	0	0	0	0	0
row 4	0	0	0	0	0
row 5	0	0	0	0	0
row 6	0	0	0	0	0
row 7	0	0	0	0	0
row 8	0	0	0	0	0
row 9	0	0	0	0	0
row 10	0	0	0	0	0
row 11	0	0	0	0	0
row 12	0	0	0	0	0
row 13	0	0	0	0	0
row 14	0	0	0	0	0
row 15	0	0	0	0	0
row 16	0	0	0	0	0
row 17	0	0	0	0	0
row 18	0	0	0	0	0
row 19	0	0	0	0	0
row 20	0	0	0	0	0
row 21	0	0	0	0	0
row 22	0	0	0	0	0
row 23	0	0	0	0	0
row 24	0	0	0	0	0
row 25	0.35044	0	0	0	0
row 26	0	0.52987	0.67274	0	0
row 27	0	-0.67274	0.52987	0	0
row 28	0	0	0	0.99938	0
row 29	0	0	0	0	0.99938

Matrix \mathbf{B}_d :

	col 1	col 2	col 3	col 4
row 1	-0.00656	0	0	0
row 2	0.00201	0	0	0
row 3	-0.00558	0	0	0
row 4	0	-0.00656	0	0
row 5	0	0.00201	0	0
row 6	0	-0.00558	0	0
row 7	0	0	-0.01832	0
row 8	0	0	-0.02324	0
row 9	0	0	-0.00418	0
row 10	0	0	0.05294	0
row 11	0	0	-0.01156	0
row 12	0	0	-0.00595	0
row 13	0	0	0.01120	0
row 14	0	0	-0.00556	0
row 15	0	0	-0.00037	0
row 16	0	0	0.00765	0
row 17	0	0	0	0.22269
row 18	0	0	0	-0.11089
row 19	0	0	0	-0.15612
row 20	0	0	0	-0.28023
row 21	0	0	0	0.09724
row 22	0	0	0	0.19131
row 23	0	0	0	0.00896
row 24	0	0	0	-0.03023
row 25	0	0	0	-0.00419
row 26	0	0	0	-0.01151
row 27	0	0	0	0.00117
row 28	0	0	0	0
row 29	0	0	0	0

Matrix \mathbf{H}^T :

	col 1	col 2
row 1	-4.00889	0
row 2	-12.98290	0
row 3	0	0
row 4	0	-4.00889
row 5	0	-12.98290
row 6	0	0
row 7	0.78246	0
row 8	-1.46497	0
row 9	-1.33571	0
row 10	0.98934	0
row 11	108.51861	0
row 12	0.64156	0
row 13	-1.13740	0
row 14	-335.82068	0
row 15	24.08479	0
row 16	-81.53566	0
row 17	0	0.26755
row 18	0	-0.45831
row 19	0	0.11705
row 20	0	0.68921
row 21	0	-0.58970
row 22	0	-0.21520
row 23	0	-0.45828
row 24	0	0.43203
row 25	0	-194.37648
row 26	0	47.36467
row 27	0	15.31794
row 28	1.00000	0
row 29	0	1.00000

A.3 LQR Model State-Space Matrices

A.3.1 Equivalent Discrete-Time

Matrix Φ :

	col 1	col 2	col 3	col 4	col 5	col 6	col 7
row 1	0.81764	0	0	0	0	0	0
row 2	0	0.98420	0.07495	0	0	0	0
row 3	0	-0.07495	0.98420	0	0	0	0
row 4	0	0	0	0.81764	0	0	0
row 5	0	0	0	0	0.98420	0.07495	0
row 6	0	0	0	0	-0.07495	0.98420	0
row 7	0	0	0	0	0	0	0.37569
row 8	0	0	0	0	0	0	-0.16762
row 9	0	0	0	0	0	0	0
row 10	0	0	0	0	0	0	0
row 11	0	0	0	0	0	0	0
row 12	0	0	0	0	0	0	0
row 13	0	0	0	0	0	0	0
row 14	0	0	0	0	0	0	0
row 15	0	0	0	0	0	0	0
row 16	0	0	0	0	0	0	0
row 17	0	0	0	0	0	0	0
row 18	0	0	0	0	0	0	0
row 19	0	0	0	0	0	0	0
row 20	0	0	0	0	0	0	0
row 21	0	0	0	0	0	0	0
row 22	0	0	0	0	0	0	0
row 23	0	0	0	0	0	0	0
row 24	0	0	0	0	0	0	0
row 25	0	0	0	0	0	0	0
row 26	0	0	0	0	0	0	0
row 27	0	0	0	0	0	0	0
row 28	0	0	0	0	0	0	0
row 29	0	0	0	0	0	0	0
row 30	-4.00889	-12.98290	0	0	0	0	0.78246
row 31	0	0	0	-4.00889	-12.98290	0	0

	col 8	col 9	col 10	col 11	col 12	col 13	col 14
row 1	0	0	0	0	0	0	0
row 2	0	0	0	0	0	0	0
row 3	0	0	0	0	0	0	0
row 4	0	0	0	0	0	0	0
row 5	0	0	0	0	0	0	0
row 6	0	0	0	0	0	0	0
row 7	0.16762	0	0	0	0	0	0
row 8	0.37569	0	0	0	0	0	0
row 9	0	-0.05613	-0.41704	0	0	0	0
row 10	0	0.41704	-0.05613	0	0	0	0
row 11	0	0	0	0.05265	0	0	0
row 12	0	0	0	0	-0.19404	0.96130	0
row 13	0	0	0	0	-0.96130	-0.19404	0
row 14	0	0	0	0	0	0	0.24555
row 15	0	0	0	0	0	0	0
row 16	0	0	0	0	0	0	0
row 17	0	0	0	0	0	0	0
row 18	0	0	0	0	0	0	0
row 19	0	0	0	0	0	0	0
row 20	0	0	0	0	0	0	0
row 21	0	0	0	0	0	0	0
row 22	0	0	0	0	0	0	0
row 23	0	0	0	0	0	0	0
row 24	0	0	0	0	0	0	0
row 25	0	0	0	0	0	0	0
row 26	0	0	0	0	0	0	0
row 27	0	0	0	0	0	0	0
row 28	0	0	0	0	0	0	0
row 29	0	0	0	0	0	0	0
row 30	-1.46497	-1.33571	0.98934	108.51861	0.64156	-1.13740	-335.82068
row 31	0	0	0	0	0	0	0

	col 15	col 16	col 17	col 18	col 19	col 20	col 21
row 1	0	0	0	0	0	0	0
row 2	0	0	0	0	0	0	0
row 3	0	0	0	0	0	0	0
row 4	0	0	0	0	0	0	0
row 5	0	0	0	0	0	0	0
row 6	0	0	0	0	0	0	0
row 7	0	0	0	0	0	0	0
row 8	0	0	0	0	0	0	0
row 9	0	0	0	0	0	0	0
row 10	0	0	0	0	0	0	0
row 11	0	0	0	0	0	0	0
row 12	0	0	0	0	0	0	0
row 13	0	0	0	0	0	0	0
row 14	0	0	0	0	0	0	0
row 15	0.50511	0.64753	0	0	0	0	0
row 16	-0.64753	0.50511	0	0	0	0	0
row 17	0	0	0.08463	0.32831	0	0	0
row 18	0	0	-0.32831	0.08463	0	0	0
row 19	0	0	0	0	0.19149	-0.05593	0
row 20	0	0	0	0	0.05593	0.19149	0
row 21	0	0	0	0	0	0	-0.28373
row 22	0	0	0	0	0	0	0.41095
row 23	0	0	0	0	0	0	0
row 24	0	0	0	0	0	0	0
row 25	0	0	0	0	0	0	0
row 26	0	0	0	0	0	0	0
row 27	0	0	0	0	0	0	0
row 28	0	0	0	0	0	0	0
row 29	0	0	0	0	0	0	0
row 30	24.08479	-81.53566	0	0	0	0	0
row 31	0	0	0.26755	-0.45831	0.11705	0.68921	-0.58970

	col 22	col 23	col 24	col 25	col 26	col 27	col 28
row 1	0	0	0	0	0	0	0
row 2	0	0	0	0	0	0	0
row 3	0	0	0	0	0	0	0
row 4	0	0	0	0	0	0	0
row 5	0	0	0	0	0	0	0
row 6	0	0	0	0	0	0	0
row 7	0	0	0	0	0	0	0
row 8	0	0	0	0	0	0	0
row 9	0	0	0	0	0	0	0
row 10	0	0	0	0	0	0	0
row 11	0	0	0	0	0	0	0
row 12	0	0	0	0	0	0	0
row 13	0	0	0	0	0	0	0
row 14	0	0	0	0	0	0	0
row 15	0	0	0	0	0	0	0
row 16	0	0	0	0	0	0	0
row 17	0	0	0	0	0	0	0
row 18	0	0	0	0	0	0	0
row 19	0	0	0	0	0	0	0
row 20	0	0	0	0	0	0	0
row 21	-0.41095	0	0	0	0	0	0
row 22	-0.28373	0	0	0	0	0	0
row 23	0	-0.15543	0.95661	0	0	0	0
row 24	0	-0.95661	-0.15543	0	0	0	0
row 25	0	0	0	0.35044	0	0	0
row 26	0	0	0	0	0.52987	0.67274	0
row 27	0	0	0	0	-0.67274	0.52987	0
row 28	0	0	0	0	0	0	0.99938
row 29	0	0	0	0	0	0	0
row 30	0	0	0	0	0	0	1.00000
row 31	-0.21520	-0.45828	0.43203	-194.37648	47.36467	15.31794	0

	col 29	col 30	col 31
row 1	0	0	0
row 2	0	0	0
row 3	0	0	0
row 4	0	0	0
row 5	0	0	0
row 6	0	0	0
row 7	0	0	0
row 8	0	0	0
row 9	0	0	0
row 10	0	0	0
row 11	0	0	0
row 12	0	0	0
row 13	0	0	0
row 14	0	0	0
row 15	0	0	0
row 16	0	0	0
row 17	0	0	0
row 18	0	0	0
row 19	0	0	0
row 20	0	0	0
row 21	0	0	0
row 22	0	0	0
row 23	0	0	0
row 24	0	0	0
row 25	0	0	0
row 26	0	0	0
row 27	0	0	0
row 28	0	0	0
row 29	0.99938	0	0
row 30	0	1.00000	0
row 31	1.00000	0	1.00000

Matrix \mathbf{B}_d :

	col 1	col 2	col 3	col 4
row 1	-0.00656	0	0	0
row 2	0.00201	0	0	0
row 3	-0.00558	0	0	0
row 4	0	-0.00656	0	0
row 5	0	0.00201	0	0
row 6	0	-0.00558	0	0
row 7	0	0	-0.01832	0
row 8	0	0	-0.02324	0
row 9	0	0	-0.00418	0
row 10	0	0	0.05294	0
row 11	0	0	-0.01156	0
row 12	0	0	-0.00595	0
row 13	0	0	0.01120	0
row 14	0	0	-0.00556	0
row 15	0	0	-0.00037	0
row 16	0	0	0.00765	0
row 17	0	0	0	0.22269
row 18	0	0	0	-0.11089
row 19	0	0	0	-0.15612
row 20	0	0	0	-0.28023
row 21	0	0	0	0.09724
row 22	0	0	0	0.19131
row 23	0	0	0	0.00896
row 24	0	0	0	-0.03023
row 25	0	0	0	-0.00419
row 26	0	0	0	-0.01151
row 27	0	0	0	0.00117
row 28	0	0	0	0
row 29	0	0	0	0
row 30	0	0	0	0
row 31	0	0	0	0

Matrix \mathbf{C}^T :

	col 1	col 2
row 1	-4.00889	0
row 2	-12.98290	0
row 3	0	0
row 4	0	-4.00889
row 5	0	-12.98290
row 6	0	0
row 7	0.78246	0
row 8	-1.46497	0
row 9	-1.33571	0
row 10	0.98934	0
row 11	108.51861	0
row 12	0.64156	0
row 13	-1.13740	0
row 14	-335.82068	0
row 15	24.08479	0
row 16	-81.53566	0
row 17	0	0.26755
row 18	0	-0.45831
row 19	0	0.11705
row 20	0	0.68921
row 21	0	-0.58970
row 22	0	-0.21520
row 23	0	-0.45828
row 24	0	0.43203
row 25	0	-194.37648
row 26	0	47.36467
row 27	0	15.31794
row 28	1.00000	0
row 29	0	1.00000
row 30	0	0
row 31	0	0

A.4 LTR-Based State Weighting Matrices

Matrix \mathbf{X} (based on $q_{LTR} = 1E3$):

	col 1	col 2	col 3	col 4	col 5	col 6
row 1	16071.23	52047.07	0	0	0	0
row 2	52047.07	168555.70	0	0	0	0
row 3	0	0	0	0	0	0
row 4	0	0	0	16071.23	52047.07	0
row 5	0	0	0	52047.07	168555.70	0
row 6	0	0	0	0	0	0
row 7	-3136.80	-10158.59	0	0	0	0
row 8	5872.91	19019.55	0	0	0	0
row 9	5354.73	17341.41	0	0	0	0
row 10	-3966.15	-12844.46	0	0	0	0
row 11	-435039.60	-1408886.33	0	0	0	0
row 12	-2571.95	-8329.31	0	0	0	0
row 13	4559.70	14766.69	0	0	0	0
row 14	1346269.46	4359926.40	0	0	0	0
row 15	-96553.35	-312690.38	0	0	0	0
row 16	326867.81	1058569.34	0	0	0	0
row 17	0	0	0	-1072.56	-3473.52	0
row 18	0	0	0	1837.31	5950.19	0
row 19	0	0	0	-469.26	-1519.71	0
row 20	0	0	0	-2762.99	-8948.00	0
row 21	0	0	0	2364.03	7655.96	0
row 22	0	0	0	862.71	2793.91	0
row 23	0	0	0	1837.18	5949.76	0
row 24	0	0	0	-1731.97	-5609.01	0
row 25	0	0	0	779234.70	2523570.53	0
row 26	0	0	0	-189879.95	-614930.84	0
row 27	0	0	0	-61407.98	-198871.25	0
row 28	-4008.89	-12982.90	0	0	0	0
row 29	0	0	0	-4008.89	-12982.90	0
row 30	0	0	0	0	0	0
row 31	0	0	0	0	0	0

	col 7	col 8	col 9	col 10	col 11	col 12
row 1	-3136.80	5872.91	5354.73	-3966.15	-435039.60	-2571.95
row 2	-10158.59	19019.55	17341.41	-12844.46	-1408886.33	-8329.31
row 3	0	0	0	0	0	0
row 4	0	0	0	0	0	0
row 5	0	0	0	0	0	0
row 6	0	0	0	0	0	0
row 7	612.24	-1146.28	-1045.14	774.12	84911.40	501.99
row 8	-1146.28	2146.14	1956.78	-1449.35	-158976.44	-939.87
row 9	-1045.14	1956.78	1784.13	-1321.47	-144949.56	-856.94
row 10	774.12	-1449.35	-1321.47	978.79	107361.48	634.72
row 11	84911.40	-158976.44	-144949.56	107361.48	11776289.00	69621.24
row 12	501.99	-939.87	-856.94	634.72	69621.24	411.60
row 13	-889.97	1666.25	1519.23	-1125.27	-123428.60	-729.71
row 14	-262766.02	491967.00	448559.54	-332239.84	-36442793.45	-215449.24
row 15	18845.37	-35283.47	-32170.33	23827.97	2613647.54	15451.84
row 16	-63798.34	119447.24	108908.12	-80666.25	-8848136.48	-52310.05
row 17	0	0	0	0	0	0
row 18	0	0	0	0	0	0
row 19	0	0	0	0	0	0
row 20	0	0	0	0	0	0
row 21	0	0	0	0	0	0
row 22	0	0	0	0	0	0
row 23	0	0	0	0	0	0
row 24	0	0	0	0	0	0
row 25	0	0	0	0	0	0
row 26	0	0	0	0	0	0
row 27	0	0	0	0	0	0
row 28	782.46	-1464.97	-1335.71	989.34	108518.61	641.56
row 29	0	0	0	0	0	0
row 30	0	0	0	0	0	0
row 31	0	0	0	0	0	0

	col 13	col 14	col 15	col 16	col 17	col 18
row 1	4559.70	1346269.46	-96553.35	326867.81	0	0
row 2	14766.69	4359926.40	-312690.38	1058569.34	0	0
row 3	0	0	0	0	0	0
row 4	0	0	0	0	-1072.56	1837.31
row 5	0	0	0	0	-3473.52	5950.19
row 6	0	0	0	0	0	0
row 7	-889.97	-262766.02	18845.37	-63798.34	0	0
row 8	1666.25	491967.00	-35283.47	119447.24	0	0
row 9	1519.23	448559.54	-32170.33	108908.12	0	0
row 10	-1125.27	-332239.84	23827.97	-80666.25	0	0
row 11	-123428.60	-36442793.45	2613647.54	-8848136.48	0	0
row 12	-729.71	-215449.24	15451.84	-52310.05	0	0
row 13	1293.67	381960.99	-27393.93	92738.31	0	0
row 14	381960.99	112775526.65	-8088169.15	27381360.13	0	0
row 15	-27393.93	-8088169.15	580076.92	-1963768.90	0	0
row 16	92738.31	27381360.13	-1963768.90	6648063.68	0	0
row 17	0	0	0	0	71.58	-122.62
row 18	0	0	0	0	-122.62	210.05
row 19	0	0	0	0	31.32	-53.65
row 20	0	0	0	0	184.40	-315.87
row 21	0	0	0	0	-157.77	270.26
row 22	0	0	0	0	-57.58	98.63
row 23	0	0	0	0	-122.61	210.03
row 24	0	0	0	0	115.59	-198.00
row 25	0	0	0	0	-52004.66	89084.61
row 26	0	0	0	0	12672.23	-21707.69
row 27	0	0	0	0	4098.25	-7020.36
row 28	-1137.40	-335820.68	24084.79	-81535.66	0	0
row 29	0	0	0	0	267.55	-458.31
row 30	0	0	0	0	0	0
row 31	0	0	0	0	0	0

	col 19	col 20	col 21	col 22	col 23	col 24
row 1	0	0	0	0	0	0
row 2	0	0	0	0	0	0
row 3	0	0	0	0	0	0
row 4	-469.26	-2762.99	2364.03	862.71	1837.18	-1731.97
row 5	-1519.71	-8948.00	7655.96	2793.91	5949.76	-5609.01
row 6	0	0	0	0	0	0
row 7	0	0	0	0	0	0
row 8	0	0	0	0	0	0
row 9	0	0	0	0	0	0
row 10	0	0	0	0	0	0
row 11	0	0	0	0	0	0
row 12	0	0	0	0	0	0
row 13	0	0	0	0	0	0
row 14	0	0	0	0	0	0
row 15	0	0	0	0	0	0
row 16	0	0	0	0	0	0
row 17	31.32	184.40	-157.77	-57.58	-122.61	115.59
row 18	-53.65	-315.87	270.26	98.63	210.03	-198.00
row 19	13.70	80.68	-69.03	-25.19	-53.64	50.57
row 20	80.68	475.02	-406.43	-148.32	-315.85	297.76
row 21	-69.03	-406.43	347.74	126.90	270.24	-254.77
row 22	-25.19	-148.32	126.90	46.31	98.62	-92.97
row 23	-53.64	-315.85	270.24	98.62	210.02	-197.99
row 24	50.57	297.76	-254.77	-92.97	-197.99	186.65
row 25	-22752.62	-133967.11	114623.01	41829.65	89078.14	-83976.62
row 26	5544.24	32644.43	-27930.75	-10192.84	-21706.11	20463.00
row 27	1793.03	10557.35	-9032.92	-3296.41	-7019.85	6617.82
row 28	0	0	0	0	0	0
row 29	117.05	689.21	-589.70	-215.20	-458.28	432.03
row 30	0	0	0	0	0	0
row 31	0	0	0	0	0	0

	col 25	col 26	col 27	col 28	col 29	col 30	col 31
row 1	0	0	0	-4008.89	0	0	0
row 2	0	0	0	-12982.90	0	0	0
row 3	0	0	0	0	0	0	0
row 4	779234.70	-189879.95	-61407.98	0	-4008.89	0	0
row 5	2523570.53	-614930.84	-198871.25	0	-12982.90	0	0
row 6	0	0	0	0	0	0	0
row 7	0	0	0	782.46	0	0	0
row 8	0	0	0	-1464.97	0	0	0
row 9	0	0	0	-1335.71	0	0	0
row 10	0	0	0	989.34	0	0	0
row 11	0	0	0	108518.61	0	0	0
row 12	0	0	0	641.56	0	0	0
row 13	0	0	0	-1137.40	0	0	0
row 14	0	0	0	-335820.68	0	0	0
row 15	0	0	0	24084.79	0	0	0
row 16	0	0	0	-81535.66	0	0	0
row 17	-52004.66	12672.23	4098.25	0	267.55	0	0
row 18	89084.61	-21707.69	-7020.36	0	-458.31	0	0
row 19	-22752.62	5544.24	1793.03	0	117.05	0	0
row 20	-133967.11	32644.43	10557.35	0	689.21	0	0
row 21	114623.01	-27930.75	-9032.92	0	-589.70	0	0
row 22	41829.65	-10192.84	-3296.41	0	-215.20	0	0
row 23	89078.14	-21706.11	-7019.85	0	-458.28	0	0
row 24	-83976.62	20463.00	6617.82	0	432.03	0	0
row 25	37782217.18	-9206578.65	-2977446.67	0	-194376.48	0	0
row 26	-9206578.65	2243412.29	725529.07	0	47364.67	0	0
row 27	-2977446.67	725529.07	234639.19	0	15317.94	0	0
row 28	0	0	0	1000	0	0	0
row 29	-194376.48	47364.67	15317.94	0	1000	0	0
row 30	0	0	0	0	0	10	0
row 31	0	0	0	0	0	0	10

Matrix **X** (based on $q_{LTR} = 500$):

	col 1	col 2	col 3	col 4	col 5	col 6
row 1	8035.62	26023.54	0	0	0	0
row 2	26023.54	84277.85	0	0	0	0
row 3	0	0	0	0	0	0
row 4	0	0	0	8035.62	26023.54	0
row 5	0	0	0	26023.54	84277.85	0
row 6	0	0	0	0	0	0
row 7	-1568.40	-5079.30	0	0	0	0
row 8	2936.45	9509.78	0	0	0	0
row 9	2677.36	8670.70	0	0	0	0
row 10	-1983.07	-6422.23	0	0	0	0
row 11	-217519.80	-704443.16	0	0	0	0
row 12	-1285.97	-4164.66	0	0	0	0
row 13	2279.85	7383.35	0	0	0	0
row 14	673134.73	2179963.20	0	0	0	0
row 15	-48276.68	-156345.19	0	0	0	0
row 16	163433.90	529284.67	0	0	0	0
row 17	0	0	0	-536.28	-1736.76	0
row 18	0	0	0	918.66	2975.09	0
row 19	0	0	0	-234.63	-759.85	0
row 20	0	0	0	-1381.49	-4474.00	0
row 21	0	0	0	1182.01	3827.98	0
row 22	0	0	0	431.36	1396.95	0
row 23	0	0	0	918.59	2974.88	0
row 24	0	0	0	-865.98	-2804.51	0
row 25	0	0	0	389617.35	1261785.26	0
row 26	0	0	0	-94939.98	-307465.42	0
row 27	0	0	0	-30703.99	-99435.62	0
row 28	-2004.45	-6491.45	0	0	0	0
row 29	0	0	0	-2004.45	-6491.45	0
row 30	0	0	0	0	0	0
row 31	0	0	0	0	0	0

	col 7	col 8	col 9	col 10	col 11	col 12
row 1	-1568.40	2936.45	2677.36	-1983.07	-217519.80	-1285.97
row 2	-5079.30	9509.78	8670.70	-6422.23	-704443.16	-4164.66
row 3	0	0	0	0	0	0
row 4	0	0	0	0	0	0
row 5	0	0	0	0	0	0
row 6	0	0	0	0	0	0
row 7	306.12	-573.14	-522.57	387.06	42455.70	251.00
row 8	-573.14	1073.07	978.39	-724.67	-79488.22	-469.93
row 9	-522.57	978.39	892.06	-660.73	-72474.78	-428.47
row 10	387.06	-724.67	-660.73	489.39	53680.74	317.36
row 11	42455.70	-79488.22	-72474.78	53680.74	5888144.50	34810.62
row 12	251.00	-469.93	-428.47	317.36	34810.62	205.80
row 13	-444.98	833.12	759.62	-562.63	-61714.30	-364.85
row 14	-131383.01	245983.50	224279.77	-166119.92	-18221396.73	-107724.62
row 15	9422.68	-17641.74	-16085.16	11913.99	1306823.77	7725.92
row 16	-31899.17	59723.62	54454.06	-40333.12	-4424068.24	-26155.02
row 17	0	0	0	0	0	0
row 18	0	0	0	0	0	0
row 19	0	0	0	0	0	0
row 20	0	0	0	0	0	0
row 21	0	0	0	0	0	0
row 22	0	0	0	0	0	0
row 23	0	0	0	0	0	0
row 24	0	0	0	0	0	0
row 25	0	0	0	0	0	0
row 26	0	0	0	0	0	0
row 27	0	0	0	0	0	0
row 28	391.23	-732.48	-667.86	494.67	54259.31	320.78
row 29	0	0	0	0	0	0
row 30	0	0	0	0	0	0
row 31	0	0	0	0	0	0

	col 13	col 14	col 15	col 16	col 17	col 18
row 1	2279.85	673134.73	-48276.68	163433.90	0	0
row 2	7383.35	2179963.20	-156345.19	529284.67	0	0
row 3	0	0	0	0	0	0
row 4	0	0	0	0	-536.28	918.66
row 5	0	0	0	0	-1736.76	2975.09
row 6	0	0	0	0	0	0
row 7	-444.98	-131383.01	9422.68	-31899.17	0	0
row 8	833.12	245983.50	-17641.74	59723.62	0	0
row 9	759.62	224279.77	-16085.16	54454.06	0	0
row 10	-562.63	-166119.92	11913.99	-40333.12	0	0
row 11	-61714.30	-18221396.73	1306823.77	-4424068.24	0	0
row 12	-364.85	-107724.62	7725.92	-26155.02	0	0
row 13	646.83	190980.49	-13696.97	46369.15	0	0
row 14	190980.49	56387763.33	-4044084.58	13690680.06	0	0
row 15	-13696.97	-4044084.58	290038.46	-981884.45	0	0
row 16	46369.15	13690680.06	-981884.45	3324031.84	0	0
row 17	0	0	0	0	35.79	-61.31
row 18	0	0	0	0	-61.31	105.02
row 19	0	0	0	0	15.66	-26.82
row 20	0	0	0	0	92.20	-157.94
row 21	0	0	0	0	-78.89	135.13
row 22	0	0	0	0	-28.79	49.31
row 23	0	0	0	0	-61.31	105.02
row 24	0	0	0	0	57.79	-99.00
row 25	0	0	0	0	-26002.33	44542.31
row 26	0	0	0	0	6336.12	-10853.84
row 27	0	0	0	0	2049.13	-3510.18
row 28	-568.70	-167910.34	12042.39	-40767.83	0	0
row 29	0	0	0	0	133.77	-229.15
row 30	0	0	0	0	0	0
row 31	0	0	0	0	0	0

	col 19	col 20	col 21	col 22	col 23	col 24
row 1	0	0	0	0	0	0
row 2	0	0	0	0	0	0
row 3	0	0	0	0	0	0
row 4	-234.63	-1381.49	1182.01	431.36	918.59	-865.98
row 5	-759.85	-4474.00	3827.98	1396.95	2974.88	-2804.51
row 6	0	0	0	0	0	0
row 7	0	0	0	0	0	0
row 8	0	0	0	0	0	0
row 9	0	0	0	0	0	0
row 10	0	0	0	0	0	0
row 11	0	0	0	0	0	0
row 12	0	0	0	0	0	0
row 13	0	0	0	0	0	0
row 14	0	0	0	0	0	0
row 15	0	0	0	0	0	0
row 16	0	0	0	0	0	0
row 17	15.66	92.20	-78.89	-28.79	-61.31	57.79
row 18	-26.82	-157.94	135.13	49.31	105.02	-99.00
row 19	6.85	40.34	-34.51	-12.60	-26.82	25.29
row 20	40.34	237.51	-203.21	-74.16	-157.93	148.88
row 21	-34.51	-203.21	173.87	63.45	135.12	-127.38
row 22	-12.60	-74.16	63.45	23.16	49.31	-46.49
row 23	-26.82	-157.93	135.12	49.31	105.01	-98.99
row 24	25.29	148.88	-127.38	-46.49	-98.99	93.33
row 25	-11376.31	-66983.56	57311.50	20914.83	44539.07	-41988.31
row 26	2772.12	16322.21	-13965.38	-5096.42	-10853.05	10231.50
row 27	896.52	5278.67	-4516.46	-1648.20	-3509.92	3308.91
row 28	0	0	0	0	0	0
row 29	58.53	344.61	-294.85	-107.60	-229.14	216.02
row 30	0	0	0	0	0	0
row 31	0	0	0	0	0	0

	col 25	col 26	col 27	col 28	col 29	col 30	col 31
row 1	0	0	0	-2004.45	0	0	0
row 2	0	0	0	-6491.45	0	0	0
row 3	0	0	0	0	0	0	0
row 4	389617.35	-94939.98	-30703.99	0	-2004.45	0	0
row 5	1261785.26	-307465.42	-99435.62	0	-6491.45	0	0
row 6	0	0	0	0	0	0	0
row 7	0	0	0	391.23	0	0	0
row 8	0	0	0	-732.48	0	0	0
row 9	0	0	0	-667.86	0	0	0
row 10	0	0	0	494.67	0	0	0
row 11	0	0	0	54259.31	0	0	0
row 12	0	0	0	320.78	0	0	0
row 13	0	0	0	-568.70	0	0	0
row 14	0	0	0	-167910.34	0	0	0
row 15	0	0	0	12042.39	0	0	0
row 16	0	0	0	-40767.83	0	0	0
row 17	-26002.33	6336.12	2049.13	0	133.77	0	0
row 18	44542.31	-10853.84	-3510.18	0	-229.15	0	0
row 19	-11376.31	2772.12	896.52	0	58.53	0	0
row 20	-66983.56	16322.21	5278.67	0	344.61	0	0
row 21	57311.50	-13965.38	-4516.46	0	-294.85	0	0
row 22	20914.83	-5096.42	-1648.20	0	-107.60	0	0
row 23	44539.07	-10853.05	-3509.92	0	-229.14	0	0
row 24	-41988.31	10231.50	3308.91	0	216.02	0	0
row 25	18891108.59	-4603289.33	-1488723.33	0	-97188.24	0	0
row 26	-4603289.33	1121706.15	362764.54	0	23682.34	0	0
row 27	-1488723.33	362764.54	117319.59	0	7658.97	0	0
row 28	0	0	0	500	0	0	0
row 29	-97188.24	23682.34	7658.97	0	500	0	0
row 30	0	0	0	0	0	10	0
row 31	0	0	0	0	0	0	10

Appendix B. MATLAB[®] Code

B.1 Tilt and Tip Simulation

```
% *****
% This code develops time-correlated Phase Screens using the
% Fourier Series method. Angular Tilt and Tip is then extracted
% from each Phase Screen.

% The code is provided by Lt Col Matthew Goda, PhD, Air Force Institute
% of Technology (AFIT).

% It has been modified by Capt Neil Paris, AFIT, for thesis work.

% 19 Feb 06
% *****

tic
clear;clc;
wind = 22; % wind speed in meters/sec
xy_wind = sqrt(10^2 / 2); % x-dir and y-dir wind component
sample = 5000; % sampling rate in Hz
N = 65; % Number of pixels
d = 3.5; % Diameter of aperture (meters)
xy0 = linspace(0,d,N); % desired initial position of phase screen
y0 = xy0;
dxy = abs(xy0(1) - xy0(2)); % x and y pixel spacing
r0=0.05;
l0=0.0001;
L0=100;
lambda = .5e-6;
run_time = 4;
num_its = run_time*sample;

translate = repmat((xy_wind/sample)*[0:num_its-1]',1,N);
xy_temp = repmat(xy0,num_its,1);
x = translate + xy_temp;
y = x;

% Find tip and tilt Zernike polynomials
xa = repmat(linspace(-1,1,N),N,1);
ya = repmat(linspace(-1,1,N)',1,N);
[theta r] = cart2pol(xa,ya);
n = 1; m = 1;
tilt_zern = sqrt(2*(n+1)/pi)*r.*cos(m*theta)/(d/2);
```

```

tilt_zern(:,ceil(N/2)) = 0;
tip_zern = sqrt(2*(n+1)/pi)*r.*sin(m*theta)/(d/2);
tip_zern(ceil(N/2),:) = 0;
tilt_zern = tilt_zern .* ((xa.^2+ya.^2)<=1);
tip_zern = tip_zern .* ((xa.^2+ya.^2)<=1);

clear xy0 translate xy_temp xa ya theta r

[kx ky sqrt_psd] = get_RFS_PSD(dxy,r0,l0,L0);

fid_tilt1 = fopen('tilt1','w');
fid_tilt2 = fopen('tilt2','w');
fid_tip1 = fopen('tip1','w');
fid_tip2 = fopen('tip2','w');
fid_tilt_ang1 = fopen('tilt_ang1','w');
fid_tilt_ang2 = fopen('tilt_ang2','w');
fid_tip_ang1 = fopen('tip_ang1','w');
fid_tip_ang2 = fopen('tip_ang2','w');

seed = 167984;
randn('seed',seed)
rand_weights = randn(1,1,length(kx)) + j*randn(1,1,length(kx));

Dphi = zeros(1,N-1);

ten=num_its/10;
for ii=1:num_its

    %----- Make the Kernel -----
    kern = make_polar_kern(kx,ky,x(ii,:),y(ii,:),sqrt_psd);
    %-----

    %----- Make Phase Screen -----
    [screen1(:,:),screen2(:,:)] = make_polar_screen(kern,rand_weights);
    %-----

    %----- Extract the Tip and Tilt Phase -----
    a_tilt1 = sum(sum(screen1.*tilt_zern))*dxy*dxy;
    a_tilt2 = sum(sum(screen2.*tilt_zern))*dxy*dxy;
    a_tip1 = sum(sum(screen1.*tip_zern))*dxy*dxy;
    a_tip2 = sum(sum(screen2.*tip_zern))*dxy*dxy;
    tilt1 = (a_tilt1*tilt_zern);
    tilt2 = (a_tilt2*tilt_zern);
    tip1 = (a_tip1*tip_zern);

```



```

tip2 = (a_tip2*tip_zern);
%-----

% ----- Convert the Tip and Tilt Phase into Angles -----
tilt1_phase = tilt1(ceil(N/2),N);
tilt1_waves = tilt1_phase/(2*pi);
tilt1_angle = tilt1_waves*lambda/(d/2);
tilt2_phase = tilt2(ceil(N/2),N);
tilt2_waves = tilt2_phase/(2*pi);
tilt2_angle = tilt2_waves*lambda/(d/2);
tip1_phase = tip1(N,ceil(N/2));
tip1_waves = tip1_phase/(2*pi);
tip1_angle = tip1_waves*lambda/(d/2);
tip2_phase = tip2(N,ceil(N/2));
tip2_waves = tip2_phase/(2*pi);
tip2_angle = tip2_waves*lambda/(d/2);
%-----

fwrite(fid_tilt1,tilt1_phase,'real*8');
fwrite(fid_tilt2,tilt2_phase,'real*8');
fwrite(fid_tip1,tip1_phase,'real*8');
fwrite(fid_tip2,tip2_phase,'real*8');

fwrite(fid_tilt_ang1,tilt1_angle,'real*8');
fwrite(fid_tilt_ang2,tilt2_angle,'real*8');
fwrite(fid_tip_ang1,tip1_angle,'real*8');
fwrite(fid_tip_ang2,tip2_angle,'real*8');

Dphi = Dphi + calc_str_fcn(screen1,'rect') +...
        calc_str_fcn(screen2,'rect');

if mod(ii,ten)==0
    percent = (ii/num_its)*100;
    disp([num2str(percent) '%']);
end
end

Dphi = Dphi/2/num_its;

fclose('all');

t = toc

save ('time', 't')
save ('Specs', 'N', 'sample', 'run_time', 'wind', 'r0', 'seed')

```

```

save ('Struct_func', 'Dphi')

*****
function [kx ky sqrt_psd] = get_RFS_PSD(dx,r0,l0,L0)

Q = 1.5;    % Factor to set number of points
max_iter = 1000;    % num of iterations for midpoint selection
tol = 1e-12;    % tolerance for midpoint selection

kappa0 = 1/L0;
kappam1 = 5.92/l0;
% set max frequency also based on sampling ...
% need max freq < 1/(2*dx), which is a period evry 2 samples
kappam2 = 2*pi / (2*dx);
kappam = min(kappam1, kappam2);

k_max = kappam;
k_min = 2*pi/5/L0; % half the lowest frequency demanded by the outer scale

log_k_min = log10(k_min);
log_k_max = log10(k_max);
delta_log = log_k_max - log_k_min;
num_freq_pts = ceil(delta_log/log10(Q));
log_k_pts = linspace(log_k_min,log_k_max,num_freq_pts);
k_pts = 10.^log_k_pts;    % These are the radial frequency values used

k_low  = k_pts(1:end - 1);
k_low2 = (k_pts(1:end - 1)).^2;
k_high = k_pts(2:end);
k_high2 = (k_pts(2:end)).^2;
delta_k = diff(k_pts);

% need to find the 2D power in each annular segment from PSD
d_phi2 = zeros(1,length(delta_k));
for i1 = 1:length(delta_k)
    d_phi2(i1) = int_vK2D2(r0,l0,L0,k_pts(i1),k_pts(i1 + 1));
end

%%%%%%%%%%%% loop to pick point to use for sinusoids
% define differential area
k_area = pi*(k_high2 - k_low2);
% provide a guess at the desired midpoint and then iterate to choose best

```

```

% midpoint as defined by place where power in annular segment is equal to
% product of area and PSD at that midpoint
k_mid = 0.5*delta_k + k_low;
cnt = 0;

for index = 1:length(k_mid)
    % seed binary search with endpoints and midpoint
    k_lower = k_pts(index);
    k_upper = k_pts(index + 1);
    B = k_mid(index);
    phi_mid = vK_phi(B^2,r0,l0,L0);
    ip = phi_mid*k_area(index);
    while (cnt <= max_iter) && (abs(ip - d_phi2(index)) > tol)
        if ip < d_phi2(index)
            k_upper = B;
            B = k_lower + .5*(B - k_lower);
        else
            k_lower = B;
            B = k_lower + .5*(k_upper - B);
        end
        phi_mid = vK_phi(B^2,r0,l0,L0);
        ip = phi_mid*k_area(index);
        cnt = cnt + 1;
        if cnt == max_iter & abs(ip - d_phi2(index)) > tol*100
            disp('Warning maximum iterations reached in kappa calc');
            disp(['          tol: ' num2str(abs(ip - d_phi2(index)))])
        end
    end
    k_mid(index) = B;
    cnt = 0;
end

% this can be changed later to include a progression of theta slices
num_theta = 32; % number of slices in theta
theta = linspace(pi/num_theta,2*pi - pi/num_theta,num_theta);
theta = theta-theta(1);
theta = repmat(theta,[1,length(k_mid)]);

k = [];
d_area = [];
for i1 = 1:length(k_mid)
    k = [k repmat(k_mid(i1),[1,num_theta])];
    d_area = [d_area repmat(k_area(i1)/num_theta,[1,num_theta])];
end

```

```

kx = k.*cos(theta); % cartesian points for given theta value
ky = k.*sin(theta);

sqrt_phi = sqrt(vK_phi(k.^2,r0,l0,L0));
sqrt_area = sqrt(d_area);
sqrt_psd = sqrt_area .* sqrt_phi;

return

*****
function phi = int_vK2D(r0,l0,L0,kappa_min,kappa_max)
% kappa is assumed to be in units of rad/m in vK_phi.m
% note that kappa_max should be in units of rad/m

kappa = logspace(log10(kappa_min),log10(kappa_max),1000);
kappa_low = kappa(1:end - 1);
kappa_high = kappa(2:end);
kappa_low2 = kappa_low.^2;
kappa_high2 = kappa_high.^2;

phi_low = vK_phi(kappa_low2,r0,l0,L0);
phi_high = vK_phi(kappa_high2,r0,l0,L0);
height = phi_low - phi_high;

% adding cylinders with partial cones at top (trapezoidal integral)
phi = sum(...
    pi/3*height.*(kappa_high2 + kappa_high.*kappa_low - 2*kappa_low2) + ...
    pi*phi_high.*(kappa_high2 - kappa_low2) );
return

*****
% calculate the von Karman phase spectrum for a given kappa, r0, l0, and L0
function PHI = vK_phi(kappa2,r0,l0,L0)

PHI = 0.4916693*r0^(-5/3)*exp(-kappa2./(5.92/l0).^2).*...
    (kappa2 + (1/L0).^2).^(-11/6);
return

*****
function [kern] = make_polar_ker(kx,ky,x,y,sqrt_psd)

Nx = length(x);
Ny = length(y);

```

```

for i1 = 1:Nx
    for i2 = 1:Ny
        for i3 = 1:length(kx)
            kern(i2,i1,i3) = sqrt_psd(i3)*...
                exp(j*(x(i1)*kx(i3) + y(i2)*ky(i3)));
        end
    end
end

*****
%=====
% Creates a pair of phase screens with subharmonics sampled over a pseudo-
% polar grid.
%=====
function [phi1,phi2] = make_polar_screen(kern, comp)

Ny = size(kern,1);
Nx = size(kern,2);
phi = zeros(Ny,Nx);

for i1 = 1:Nx
    for i2 = 1:Ny
        phi(i2,i1) = phi(i2,i1) + sum(comp.*kern(i2,i1,:),3);
    end
end

phi1 = real(phi);
phi1 = phi1-mean(mean(phi1));
phi2 = imag(phi);
phi2 = phi2-mean(mean(phi2));

return

*****
function [out, out_diag] = calc_str_fcn(phz, type)
% function to calculate the phase structure function of the
% input phase array phz. This will calculate a slice of the 2-d structure
% function average along horiz/vertical direction and also along the two
% perpendicular diagonals.
%
% use type='rect' to use full phz matrix, 'circ' will take rectangle
%           circumscribed within the assumed circular aperture
% Note output is in terms of pixels, you must figure out the relationship
% to actual physical distance, i.e. meters

```

```

if nargin<2 | ~(strcmp(type,'rect') | strcmp(type,'circ'))
    error('Need to specify rect/circ in calc_str_fcn');
end

if strcmp(type,'circ') % Take circumscribed rectangle
    l = length(phz);
    i1 = ceil(l/2*(1-1/sqrt(2)))+1;
    i2 = floor(l/2*(1+1/sqrt(2)))+1;
    phz2 = phz(i1:i2,i1:i2);
else % use whole thing
    phz2=phz;
end

sz = length(phz2);
% along horiz and vert axes
for ii = 2:sz
    temp = phz2(ii:end,:) - phz2(1:(sz+1-ii),:);
    out1(ii-1) = mean(temp(:).^2);
    temp = phz2(:,ii:end) - phz2(:,1:(sz+1-ii));
    out2(ii-1) = mean(temp(:).^2);
end
out = (out1+out2)/2;

% if requested, along diagonals
if nargout>1
    for ii = 2:sz
        temp = phz2(ii:end,ii:end) - phz2(1:(sz+1-ii),1:(sz+1-ii));
        out1(ii-1) = mean(temp(:).^2);
        temp = phz2(ii:end,1:(sz+1-ii)) - phz2(1:(sz+1-ii),ii:end);
        out2(ii-1) = mean(temp(:).^2);
    end
    out_diag = (out1+out2)/2;
end
end

```

B.2 Pyramid Sensor Noise Characterization

```
% *****
% This code determines the variance (power) of the pyramid sensor
% assuming white Gaussian noise.

% The code is provided by Dr James Brown, Starfire Optical Range,
% Air Force Research Laboratory.

% It has been modified by Capt Neil Paris, AFIT, for thesis work.

% 19 Feb 06
% *****

save = 1;

azScale = 11.1; % nrad/count (scaling for CSM)
elScale = 9.67; % nrad/count

% AZIMUTH AXIS -----
d = load ('G:\Thesis\forAFIT\From3Nov\CSM_AZ_0200');
%
x = d.CSM_AZ_0200(:,1);      % CSM Command (counts) - repeated entries
y = d.CSM_AZ_0200(:,5);      % Pyramid quad-cell calculation (tilt indication)
c = sum(d.CSM_AZ_0200(:, 7:10), 2); % Pyramid Signal Counts
%
x = x - mean(x);      % Amount from mean - pos means amount above mean
                        % - neg means amount below mean
                        % This normalizes the tilt about the initial tilt angle
[xx, yy, vy] = avg_data(x, y, 0.01); % xx is sorted x without repeats
                        % yy is avg y value for each x repeat
                        % vy is the variance of the y values
                        % that make up each yy value

xx = xx*azScale;
x = x*azScale;

idx = find(abs(yy) <= 0.2);
xTemp = xx(idx);
yTemp = yy(idx);
p = polyfit(xTemp, yTemp, 1); % p = mx + b
m = p(1);                    % slope of p
yy = yy - p(2);              % Center yy about zero
y = y - p(2);

% Determine noise strength assuming it is white gaussian
Var_y_az = mean(vy(idx))/m^2; % Variance in nrad^2
```

```

                                % Same as finding the variance
                                % of all of the y values together

figure(1)
plot(x, y, '.b') % Plot of raw data(besides x-mean(x)
hold on
plot(xx, yy, '-k', 'linewidth', 4); % Plot of the mean of the data
plot([-1/m, 1/m], [-1 1], '--k', 'linewidth', 2)
hold off
set(gca,'fontsize',15)
grid on
xlabel('Commanded Tilt (nrad)');ylabel('Pyramid Tilt')
xlim([min(x) max(x)])
ylim([-0.75 0.75])
title(sprintf('Azimuth Axis -- Linear Region Slope = %0.2E per nrad', m))
legend('Output Sample','Mean Output','Linear Fit','location','northwest')
if save==1
print('-depsc2','-tiff','-f1','G:\Thesis\Thesis Plots\sensor_az');
print('-depsc2','-tiff','-f1',...
      'D:\Paris_Thesis\Thesis\Thesis Plots\sensor_az');
end
% -----

% ELEVATION AXIS -----
d = load ('G:\Thesis\forAFIT\From3Nov\CSM_EL_0200');
%
x = d.CSM_EL_0200(:,2);
y = d.CSM_EL_0200(:,6);
c = sum(d.CSM_EL_0200(:, 7:10), 2);
%
x = x - mean(x);
[xx, yy, vy] = avg_data(x, y, 0.01);
xx = xx*elScale;
x = x*elScale;
%
idx = find((yy >= -0.1) & (yy <= 0.4));
xTemp = xx(idx);
yTemp = yy(idx);
p = polyfit(xTemp, yTemp, 1);
m = p(1);
yy = yy - p(2);

% Determine noise strength assuming it is white gaussian
Var_y_el = mean(vy(idx))/m^2; % Variance in nrad^2

```



```

figure(2)
plot(x-200, y-.15, '.b') % Plot of raw data(besides x-mean(x))
hold on
plot(xx-200, yy-.15, '-k', 'linewidth', 4); % Plot of the mean of the data
plot([-1/m, 1/m], [-1 1], '--k', 'linewidth', 2)
hold off
set(gca,'fontsize',15)
grid on
xlabel('Commanded Tip (nrad)');ylabel('Pyramid Tip')
xlim([- (max(x)-200) max(x)-200])
% xlim([-1934 1934])
ylim([-0.75 0.75])
title(sprintf('Elevation Axis -- Linear Region Slope = %0.2E per nrad', m))
legend('Output Sample','Mean Output','Linear Fit','location','northwest')
if save==1
print('-depsc2','-tiff','-f2','G:\Thesis\Thesis Plots\sensor_el');
print('-depsc2','-tiff','-f2',...
      'D:\Paris_Thesis\Thesis\Thesis Plots\sensor_el');
end
% -----

*****
function [xOut, yOut, v] = avg_data(x,y,thresh)
%
%
[xx,idx] = sort(x); % Sorts x in ascending order
% idx is the index of the value in x
yy = y(idx); % keeps the y values matched to the original x vals
%
dx = diff(xx); % Absolute Difference between each element of xx
ii = find(dx >= thresh); % Indices of dx values that are >= thresh
endIdx = [ii(:); length(xx)]; % Indices of xx at end of
% each set of repeated data
begIdx = [1; ii(:)+1]; % Indices of xx at beginning of each set of
% repeated data
%
Nbins = length(begIdx);
xOut = zeros(Nbins,1);
yOut = zeros(Nbins,1);
s = zeros(Nbins,1);

for i = 1:Nbins
    xOut(i) = mean(xx(begIdx(i):endIdx(i)));
    yOut(i) = mean(yy(begIdx(i):endIdx(i)));
    v(i) = var(yy(begIdx(i):endIdx(i)),1); % Same as shifting the mean

```

```
end                                     % to zero and shifting all
                                     % corresponding y values by
                                     % same amount and then finding
                                     % mean of all the y values
                                     % squared
```

B.3 Controller Design

```
% *****
% This code builds an LQG/LTR controller. The Kalman Filter model
% and LQR model are formed and the steady-state gains are
% determined. A Simulink file, sor_PI_Tracker2, uses the gains
% and models for simulation.

% Written by Capt Neil Paris, Air Force Institute of Technology

% 19 Feb 06
% *****

clear; clc;
for dd=0:0
for bb=1:3

% ----- User Interface -----
%   Save Data? Yes: sav=1; No: sav=0
        sav = 1;

%   Run Simulink model? Yes: simu=1; No: simu=0
        simu = 1;
%   Acquisition? Yes: Acq=1; No: Acq=0;
        Acq=0;
%   Delay? Yes: Delay=1; No: Delay=0
        Delay = 0;
%   Number of Delays (in samples)?
        num_delay = 10;
%   Sensor Saturation? Yes: Sat=1; No: Sat=0
        Sat = 0;
%   Measurement Noise? Yes: Noise=1; No: Noise=0
        Noise = 1;
%   Monte Carlo Runs
        MC = 10;

%   Truth Model - Low order: tru=1, High order: tru=2
        tru = 1;

%   Controller Type - LQG: cont=1, LQG/LTR: cont=2
        cont = 1;

%   Tilt Disturbance Model
%   Mult factor on Q matrix values
        qscale = 0;
        qscale2 = 0;
```

```

%      Diameter of Aperture
      d = 3.5;
%      Coherence Radius
      r0 = .05;
%      Wavelength
      lambda = .5e-6;
%      Correlation Time
      T_tilt = 0.320;      % .13 based on wind = 21 m/s
      T_tip = 0.320;      % .32 based on wind = 21 m/s

%      Filter Plant Models Tuning Values
%      Q Matrix - diagonal entries for Q:
      q = 1e0*[1e0*ones(1,3) 1e0*ones(1,3) 1e7*ones(1,10) 1e7*ones(1,11);...
              1e0*ones(1,3) 1e0*ones(1,3) 1e6*ones(1,10) 1e6*ones(1,11);...
              1e0*ones(1,3) 1e0*ones(1,3) 1e6*ones(1,10) 1e6*ones(1,11)];
      q = q(bb,:);
%      R Matrix - diagonal entries for R:
      r = [4e5 6e5; 9e4 2e5; 9e4 2e5];
      r = r(bb,:);
%      r = [1e0 1e0]; %Baseline

%      Cost Function Tuning Values
%      X Matrix - Currently coded in LQG/LTR section
      x = 1e0*[0*ones(1,3) 0*ones(1,3)...
              0*ones(1,10) 0*ones(1,11) 0 0 1e1*1 1e1*1];

%      If using LTR, set level for additional noise:
      q_ltr = [1e3; 1e3; 500];
      q_ltr = q_ltr(bb);
%      U Matrix - diagonal entries for U:
      u = [1e0 1e0 1e4 1e4; 5e3 5e3 1e4 1e4; 1e4 1e4 1e4 1e4];
      u = u(bb,:);

% ----- Measurement Data -----
if dd==0
fid_tilt = fopen('Good Data\Jan23_129_5k_1s_10_5cm_98\tilt_ang1');
fid_tip = fopen('Good Data\Jan23_129_5k_1s_10_5cm_98\tip_ang1');
specs = load('Good Data\Jan23_129_5k_1s_10_5cm_98\Specs');
end

if dd==4
fid_tilt = fopen('Good Data\Jan21_65_5k_4s_10_5cm_0\tilt_ang1');
fid_tip = fopen('Good Data\Jan21_65_5k_4s_10_5cm_0\tip_ang1');
specs = load('Good Data\Jan21_65_5k_4s_10_5cm_0\Specs');

```

```

end

if dd==1
fid_tilt = fopen('Good Data\Jan21_65_5k_4s_22_5cm_0\tilt_ang1');
fid_tip = fopen('Good Data\Jan21_65_5k_4s_22_5cm_0\tip_ang1');
specs = load('Good Data\Jan21_65_5k_4s_22_5cm_0\Specs');
end

if dd==2
fid_tilt = fopen('Good Data\Jan21_65_5k_4s_5_5cm_0\tilt_ang1');
fid_tip = fopen('Good Data\Jan21_65_5k_4s_5_5cm_0\tip_ang1');
specs = load('Good Data\Jan21_65_5k_4s_5_5cm_0\Specs');
end

sample_rate = specs.sample; % in Hz
sample_time = 1/sample_rate;

tilt = (1e9)*fread(fid_tilt,inf,'real*8');
tip = (1e9)*fread(fid_tip,inf,'real*8');
% tip = tilt;
if Acq == 0
    tilt = tilt - tilt(1);
    tip = tip - tip(1);
end

meas.signals.dimension = 2;

% Load Actual Sensor noise strengths
n1 = load('G:\Thesis\Saved_data\Sensor_noise_az.mat');
n2 = load('G:\Thesis\Saved_data\Sensor_noise_el.mat');
sensor_noise = [n1.Var_y_az n2.Var_y_el];
% sensor_noise = [7564.1 13827];
% -----

% ----- Form CSM and FSM state space matrices for Kalman Filter -----
F1 = load('G:\Thesis\Saved_data\diag_fsm_az_9_10.mat');
F2 = load('G:\Thesis\Saved_data\diag_fsm_el_9_11.mat');
FSM_az = F1.F_az;    % azimuth axis
FSM_el = F2.F_el;    % elevation axis

C = load('G:\Thesis\Saved_data\diag_csm.mat');
CSM_az = C.CSM;      % azimuth axis
CSM_el = CSM_az;      % elevation axis

% Zero matrices for Overall Plant A matrix

```

```

zero_csm_A1 = zeros(length(CSM_az.A),length(CSM_el.A)); % 1,2
zero_csm_A2 = zero_csm_A1'; % 2,1
zero_fsm_A1 = zeros(length(FSM_az.A),length(FSM_el.A)); % 3,4
zero_fsm_A2 = zero_fsm_A1'; % 4,3
zero_urA = zeros(length(CSM_az.A)+length(CSM_el.A),... % upper right
    length(FSM_az.A)+length(FSM_el.A));
zero_llA = zero_urA'; % lower left
urA = size(zero_urA); % dimensions of upper right
llA = size(zero_llA); % dimensions of lower left

% Zero matrices for Overall Plant B matrix
% Assumes that the individual plants have a B column vector
zero_csm_B1 = zeros(size(CSM_az.B,1),1); % 1,2
zero_csm_B2 = zeros(size(CSM_el.B,1),1); % 2,1
zero_fsm_B1 = zeros(size(FSM_az.B,1),1); % 3,4
zero_fsm_B2 = zeros(size(FSM_el.B,1),1); % 4,3
zero_urB1 = [zero_csm_B1 zero_csm_B1]; % upper right, 1st row
zero_urB2 = [zero_csm_B2 zero_csm_B2]; % upper right, 2nd row
zero_llB1 = [zero_fsm_B1 zero_fsm_B1]; % lower left, 1st row
zero_llB2 = [zero_fsm_B2 zero_fsm_B2]; % lower left, 2nd row

% Zero matrices for Overall Plant C matrix
% Assumes that the individual plants have a C row vector
zero_csm_C1 = zeros(1,size(CSM_el.C,2)); % 1,2
zero_csm_C2 = zeros(1,size(CSM_az.C,2)); % 2,1
zero_fsm_C1 = zeros(1,size(FSM_el.C,2)); % 1,4
zero_fsm_C2 = zeros(1,size(FSM_az.C,2)); % 2,3

% Overall Plant A matrix
A = [CSM_az.A zero_csm_A1 zero_urA(1:length(CSM_az.A),:);...
    zero_csm_A2 CSM_el.A zero_urA(length(CSM_az.A)+1:urA(1),:);...
    zero_llA(1:length(FSM_az.A),:) FSM_az.A zero_fsm_A1;...
    zero_llA(length(FSM_az.A)+1:llA(1),:) zero_fsm_A2 FSM_el.A];

% Overall Plant B matrix
B = [CSM_az.B zero_csm_B1 zero_urB1;...
    zero_csm_B2 CSM_el.B zero_urB2;...
    zero_llB1 FSM_az.B zero_fsm_B1;...
    zero_llB2 zero_fsm_B2 FSM_el.B];

% Overall Plant C matrix
C = [CSM_az.C zero_csm_C1 FSM_az.C zero_fsm_C1;...
    zero_csm_C2 CSM_el.C zero_fsm_C2 FSM_el.C];

% Overall Plant D matrix

```

```

D = 0;

% Overall Plant state matrices
Pl = ss(A,B,C,D);

% Plant Dynamics Noise
Q = diag(q);    % Process Noise
G = eye(length(Pl.A));
% -----

% ----- Create Truth Model -----
if tru == 1
    truth = Pl;
else
    load('G:\Thesis\Saved_data\fsm_az_17_18.mat');
    TF1 = ss(fsm_est2);
    load('G:\Thesis\Saved_data\fsm_el_17_18.mat');
    TF2 = ss(fsm_est2);
    load('G:\Thesis\Saved_data\diag_csm.mat');
    TC = CSM;
    truth = truth_model(TF1, TF2, TC);
end
% -----

% ----- Kalman Filter Tilt Disturbance Model -----
var_tilt = ((0.448*((d/r0)^(5/3)))/((pi)^2))*((lambda/(d/2))^2)*((1e9)^2);
var_tip = var_tilt;
Q_disturb = diag([(2*var_tilt/T_tilt)+(qscale*(1e7))...
    (2*var_tip/T_tip)+(qscale2*(1e7))]);
A_disturb = diag([-1/T_tilt -1/T_tip]);
C_disturb = eye(size(A_disturb,1));
D_disturb = 0;
G_disturb = eye(size(A_disturb,1));
% -----

% ----- Augmented Kalman Filter State Model -----
A_aug_u_zero = zeros(size(Pl.A,1),size(A_disturb,1));
A_aug_l_zero = zeros(size(A_disturb,1),size(Pl.A,2));
A_aug = [Pl.A A_aug_u_zero; A_aug_l_zero A_disturb];
B_aug_l_zero = zeros(size(A_disturb,1),size(Pl.B,2));
B_aug = [Pl.B; B_aug_l_zero];
C_aug = [Pl.C C_disturb];
D_aug = 0;
Kf_model = ss(A_aug,B_aug,C_aug,D_aug);
G_aug_u_zero = zeros(size(G,1),size(G_disturb,2));

```

```

G_aug_l_zero = zeros(size(G_disturb,1),size(G,2));
G_aug = [G G_aug_u_zero; G_aug_l_zero G_disturb];
Q_aug_u_zero = zeros(size(Q,1),size(Q_disturb,2));
Q_aug_l_zero = zeros(size(Q_disturb,1),size(Q,2));
Q_aug = [Q Q_aug_u_zero; Q_aug_l_zero Q_disturb];

% Equivalent Discrete Time Model
dKf_model = c2d(Kf_model,sample_time);
Qd_aug = Qd(Q_aug,G_aug,Kf_model.A,sample_time);
% -----

% ----- Psuedo-Integral Discrete State Model (for LQR Model)-----
Ai = [Pl.C eye(size(A_disturb,1)) eye(size(C_aug,1))];
% -----

% ----- Discrete-time LQR Model -----
A_lqr_u_zero = zeros(size(dKf_model.A,1),size(Ai,1));
A_lqr_l_zero = zeros(size(Ai,1));
A_lqr = [dKf_model.A A_lqr_u_zero; Ai];
B_lqr_l_zero = zeros(size(Ai,1),size(dKf_model.B,2));
B_lqr = [dKf_model.B; B_lqr_l_zero];
C_lqr = [dKf_model.C zeros(size(dKf_model.C,1),size(Ai,1))];
% -----

% ----- Kalman Filter - Gain (Kf), Covariance (P), (and ss model) -----
R = diag(r); % Measurement noise
% Determines the covariance and gain
[P,E,K] = dare(dKf_model.A',dKf_model.C',G_aug*Qd_aug*G_aug',R);
Kf = K'; % care gain output is transposed
% Discrete-time Kalman Filter for Simulink
Kf_model2 = ss(A_aug,[B_aug G_aug],C_aug,D_aug);
[KF Kf2 P2] = kalmd(Kf_model2,Q_aug,R*sample_time,sample_time);
% -----

% ----- LQG/LTR -----
% Determine Feedback Gain, Kc
X = diag(x); % State weighting matrix
X_ltr = X + q_ltr*C_lqr'*C_lqr; % Add q_ltr*C'*C to X for LTR
U = diag(u); % Control weighting matrix

if cont == 1
    [Kc,E2,Gc] = dare(A_lqr,B_lqr,X,U); % Determines the optimal gain
else
    [Kc,E2,Gc] = dare(A_lqr,B_lqr,X_ltr,U); % Determines the optimal gain
end

```



```

% -----

if Noise == 1
    data_rate = .1*sample_time;
else
    data_rate = -1;
end

hh = waitbar(0,'Performing Monte Carlo Runs...');
rand('state',0)
for ii=1:MC

    seed = randint(1,1,[0,99999999]);
% ----- Run Simulink Model -----
if simu == 1
    tic
    sim('sor_PI_tracker2',specs.run_time - sample_time)
    toc
end
% -----

if MC>1
% ----- Data Collection -----
Atm(:,:,ii) = Actual_tilt_tip;
c_time = counter.time;
Reg(:,:,ii) = Reg_tilt_tip;
wgn(:,:,ii) = WGN;
Z(:,:,ii) = z.signals.values;
d_time = z.time;
Kf_ests(:,:,ii) = KFests;
input(:,:,ii) = Input;
out_CSM_az(:,:,ii) = y_CSM_az;
out_CSM_el(:,:,ii) = y_CSM_el;
out_FSM_az(:,:,ii) = y_FSM_az;
out_FSM_el(:,:,ii) = y_FSM_el;
out_az(:,:,ii) = y_az;
out_el(:,:,ii) = y_el;
states_CSM_az(:,:,ii) = x_CSM_az;
states_CSM_el(:,:,ii) = x_CSM_el;
states_FSM_az(:,:,ii) = x_FSM_az;
states_FSM_el(:,:,ii) = x_FSM_el;
% -----
clear 'Actual_tilt_tip' 'Reg_tilt_tip' 'WGN' 'z' 'KFests' 'Input' 'counter'
clear 'y_CSM_az' 'y_CSM_el' 'y_FSM_az' 'y_FSM_el' 'y_az' 'y_el' 'tout'...

```

```

        'x_CSM_az' 'x_CSM_el' 'x_FSM_az' 'x_FSM_el'
end

waitbar(ii/MC)
end %end of MC runs
close(hh)

% ----- Save Data -----
if sav == 1
    time = [num2str(specs.run_time) 'sec'];
    wind = ['wind' num2str(specs.wind) '_'];
    filename = ['D:\Paris_Thesis\Results_Data\' time wind...
        'U_' num2str(u(1)) '_' num2str(u(2)) '_' num2str(u(3))...
        '_' num2str(u(4)) '_Acq' num2str(Acq) '_Sat' num2str(Sat)...
        '_Delay' num2str(Delay) '_delay' num2str(num_delay)...
        '_WGN' num2str(Noise)];
    if MC>1
        filename = [filename '_MC'];
        save(filename,'cont','qscale','T_tilt','T_tip','q','r','x',...
            'q_ltr','u','Atm','Reg','wgn','Z','Kf_ests','input',...
            'out_CSM_az','out_CSM_el','out_FSM_az','out_FSM_el',...
            'out_az','out_el','c_time','d_time','states_CSM_az',...
            'states_CSM_el','states_FSM_az','states_CSM_el','Sat',...
            'Delay','num_delay','Noise','specs');
    else
        save(filename,'cont','qscale','T_tilt','T_tip','q','r','x',...
            'q_ltr','u','Actual_tilt_tip','Reg_tilt_tip','WGN','z',...
            'KFests','Input','y_CSM_az','y_CSM_el','y_FSM_az',...
            'y_FSM_el','y_az','y_el','tout','x_CSM_az','x_CSM_el',...
            'x_FSM_az','x_FSM_el','Sat','Delay','num_delay',...
            'Noise','specs');
    end
end
% -----

*****
function Pl = truth_model(FSM_az, FSM_el, CSM)

CSM_az = CSM;    % azimuth axis
CSM_el = CSM_az;    % elevation axis

% Zero matrices for Overall Plant A matrix
zero_csm_A1 = zeros(length(CSM_az.A),length(CSM_el.A)); % 1,2
zero_csm_A2 = zero_csm_A1'; % 2,1
zero_fsm_A1 = zeros(length(FSM_az.A),length(FSM_el.A)); % 3,4

```

```

zero_fsm_A2 = zero_fsm_A1'; % 4,3
zero_urA = zeros(length(CSM_az.A)+length(CSM_el.A),... % upper right
    length(FSM_az.A)+length(FSM_el.A));
zero_llA = zero_urA'; % lower left
urA = size(zero_urA); % dimensions of upper right
llA = size(zero_llA); % dimensions of lower left

% Zero matrices for Overall Plant B matrix
% Assumes that the individual plants have a B column vector
zero_csm_B1 = zeros(size(CSM_az.B,1),1); % 1,2
zero_csm_B2 = zeros(size(CSM_el.B,1),1); % 2,1
zero_fsm_B1 = zeros(size(FSM_az.B,1),1); % 3,4
zero_fsm_B2 = zeros(size(FSM_el.B,1),1); % 4,3
zero_urB1 = [zero_csm_B1 zero_csm_B1]; % upper right, 1st row
zero_urB2 = [zero_csm_B2 zero_csm_B2]; % upper right, 2nd row
zero_llB1 = [zero_fsm_B1 zero_fsm_B1]; % lower left, 1st row
zero_llB2 = [zero_fsm_B2 zero_fsm_B2]; % lower left, 2nd row

% Zero matrices for Overall Plant C matrix
% Assumes that the individual plants have a C row vector
zero_csm_C1 = zeros(1,size(CSM_el.C,2)); % 1,2
zero_csm_C2 = zeros(1,size(CSM_az.C,2)); % 2,1
zero_fsm_C1 = zeros(1,size(FSM_el.C,2)); % 1,4
zero_fsm_C2 = zeros(1,size(FSM_az.C,2)); % 2,3

% Overall Plant A matrix
A = [CSM_az.A zero_csm_A1 zero_urA(1:length(CSM_az.A),:);...
    zero_csm_A2 CSM_el.A zero_urA(length(CSM_az.A)+1:urA(1),:);...
    zero_llA(1:length(FSM_az.A),:) FSM_az.A zero_fsm_A1;...
    zero_llA(length(FSM_az.A)+1:llA(1),:) zero_fsm_A2 FSM_el.A];

% Overall Plant B matrix
B = [CSM_az.B zero_csm_B1 zero_urB1;...
    zero_csm_B2 CSM_el.B zero_urB2;...
    zero_llB1 FSM_az.B zero_fsm_B1;...
    zero_llB2 zero_fsm_B2 FSM_el.B];

% Overall Plant C matrix
C = [CSM_az.C zero_csm_C1 FSM_az.C zero_fsm_C1;...
    zero_csm_C2 CSM_el.C zero_fsm_C2 FSM_el.C];

% Overall Plant D matrix
D = 0;

% Overall Plant state matrices

```

```
P1 = ss(A,B,C,D);
```

```
*****
function [qd] = Qd(Q,G,Phi,Ts)
% This function determines Qd which is the discretized version
%   of the process noise matrix Q from a continuous Kalman filter model
% The routine is written by Dr. Peter Maybeck, Air Force Institute of
%   Technology, Wright-Patterson Air Force Base, OH
% The routine is a modification of a portion of Matlab's old LQED function
%   to get Qd

% Reference: This routine is based on the routine DISRW.M by Franklin,
% Powell and Workman and is described on pp. 454-455 of "Digital Control
% of Dynamic Systems".

[nx,nu] = size(G);

[nq,mq] = size(Q);
if (mq ~= nq) | (nu ~= mq), error('G and Q must be consistent.');
```

end

```
% Check if Q is positive semi-definite and symmetric
if any(eig(Q) < -eps) | (norm(Q'-Q,1)/norm(Q,1) > eps)
    disp('Warning: Q is not symmetric and positive semi-definite');
end

Zf = zeros(nx);
M = [ -Phi   G*Q*G'
      Zf    Phi'   ];
phi_aug = expm(M*Ts);
phi12 = phi_aug(1:nx,nx+1:2*nx);
phi22 = phi_aug(nx+1:2*nx,nx+1:2*nx);
qd = phi22'*phi12;
qd = (qd+qd')/2;\qqquad \qqquad % Make sure Qd is symmetric
```

Bibliography

1. Air Force Research Laboratory Directed Energy Directorate, "3.5 meter Telescope Facility." http://www.de.afrl.af.mil/SOR/3_5m_telescope.htm (Nov 2005).
2. Air Force Research Laboratory Directed Energy Directorate, "Starfire Optical Range Mission." <http://www.de.afrl.af.mil/SOR/SORmission.htm> (Nov 2005).
3. Anderson, David J. *LQG Control of a Deformable Mirror Adaptive Optics System With Time-Delayed Measurements*. MS thesis, Air Force Institute of Technology, December 1991.
4. Andrews, Larry C. and Ronald L. Phillips. *Laser Beam Propagation Through Random Media*. Bellingham, WA: SPIE - The International Society for Optical Engineering, 1998.
5. Brown, James M. "Tracker Development for the Starfire Optical Range 3.5-meter Telescope Adaptive Optics System." *SPIE Vol. 4839*, edited by Peter L. Wizinowich and Domenico Bonaccini. 240–249. August 2002.
6. Brown, James M. Personal, telephone, and email interviews, February to December 2005.
7. Brown, James M. "SOR AO System: 3.5m Telescope and Coude Path." Graphic is used with author's permission from Powerpoint slides briefed by the author, January 2005.
8. Burl, Jeffrey B. *Linear Optimal Control*. Menlo Park, CA: Addison Wesley Longman, Inc., 1999.
9. Lee, David, et al. "First Results for the Starfire Optical Range 3.5-m Telescope Adaptive Optics System: Point-Spread Functions and Tracking Performance." *SPIE Vol. 3353*. 1080–1091. March 1998.
10. MathWorks, "Signal Processing Toolbox Documentation." <http://www.mathworks.com/access/helpdesk/help/toolbox/signal/spectop9.html> (Feb 24, 2006).
11. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Vol. 3*. Mathematics in Science and Engineering. Academic Press, Inc., 1982.
12. Maybeck, Peter S. *Stochastic Models, Estimation, and Control, Vol. 1*. Mathematics in Science and Engineering. Arlington, VA: Navtech Book and Software Store, 1994.
13. Reid, J. Gary. *Linear System Fundamentals: Continuous and Discrete, Classic and Modern*. New York: McGraw-Hill Inc., 1983.
14. Roggemann, Michael C. and Byron M. Welsh. *Imaging Through Turbulence*. Boca Raton, FL: CRC Press LLC, 1996.
15. Spinhirne, J. M. et al. "The Starfire Optical Range 3.5m Telescope Adaptive Optics System." *SPIE Vol. 3353*. 22–33. March 1998.

16. Spinhirne, James M. and George A. Ameer. "Adaptive Optics Using the 3.5m Starfire Optical Range Telescope." *SPIE Vol.3126*. 257–268. July 1997.
17. Tyson, Robert K. *Introduction to Adaptive Optics*. Bellingham, WA: SPIE - The International Society for Optical Engineering, 2000.
18. Von Bokern, Mark Alan. *Design of a Linear Quadratic Gaussian Control Law for an Adaptive Optics System*. MS thesis, Air Force Institute of Technology, December 1990.

Vita

Captain Neil D. Paris earned a Bachelor of Science degree in Electrical Engineering from Parks College of Saint Louis University, and was commissioned on 19 May 2001 through ROTC. His first assignment was with the 746th Test Squadron at Holloman AFB, NM, as a flight test engineer evaluating aircraft and weapon navigation systems. In May 2004, he was assigned to the Air Force Institute of Technology, Wright-Patterson AFB, OH, to earn a Master of Science degree in Electrical Engineering. Capt Paris' follow-on assignment is to the Starfire Optical Range at Kirtland AFB, NM.

Permanent address: Louisville, KY, USA

REPORT DOCUMENTATION PAGE					<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.						
1. REPORT DATE (DD-MM-YYYY) 23-03-2006		2. REPORT TYPE Master's Thesis			3. DATES COVERED (From — To) Jun 2004 — Mar 2006	
4. TITLE AND SUBTITLE <div style="text-align: center;">LQG/LTR Tilt and Tip Control for the Starfire Optical Range 3.5-meter Telescope's Adaptive Optics System</div>				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Neil D. Paris, Capt, USAF				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765					8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/06-45	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Dr. James Brown AFMC/AFRL/DES 3550 Aberdeen Ave. SE Kirtland Air Force Base, NM 87117 Phone: 505-846-5836, Email: james.brown@kirtland.af.mil					10. SPONSOR/MONITOR'S ACRONYM(S)	
					11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT The Air Force Research Laboratory has sponsored research on the tracking control loop portion of the adaptive optics system in the Starfire Optical Range 3.5-meter telescope at Kirtland Air Force Base. The control loop includes two steering mirrors (Coarse Steering Mirror and Fine Steering Mirror) used to remove wavefront tilt and tip phase distortion from light entering the telescope. The objective of this research is to design a single Linear Quadratic Gaussian controller to control both steering mirrors in order to eliminate wavefront tilt and tip distortions induced by the earth's atmosphere, and to evaluate the stability robustness and performance of the controller through simulation and Monte Carlo analysis. Controller design elements and simulation parameters are varied to examine and compare resulting performance and robustness characteristics. The controller design is limited to the use of linear models even though components within the control loop have some nonlinear characteristics; however, consideration has been given to the nonlinear aspects of the system via the simulation environment in order to observe the linear controller in a near-real-world environment.						
15. SUBJECT TERMS Adaptive Control Systems, Tracking Telescopes, Linear Quadratic Gaussian, Adaptive Optics, Mirrors						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <div style="text-align: center;">UU</div>		18. NUMBER OF PAGES <div style="text-align: center;">252</div>	
a. REPORT	b. ABSTRACT	c. THIS PAGE				
U	U	U	19a. NAME OF RESPONSIBLE PERSON Juan R. Vasquez, Lt Col, USAF			
				19b. TELEPHONE NUMBER (include area code) (937) 255-3636 x7231, juan.vasquez@afit.edu		